

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A242 537



DTIC
ELECTE
NOV 18 1991
S D D

THESIS

EFFICIENT TECHNIQUE FOR
CALCULATING NEAR-FIELDS
DUE TO 2-D SOURCES

by

Richard A. Rostant

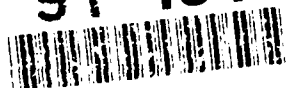
December, 1990

Thesis Advisor:

Ramakrishna Janaswamy

Approved for public release; distribution is unlimited.

91-15190



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No 0704-0188	
1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
4. PERFORMING ORGANIZATION REPORT NUMBER(S)					
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b OFFICE SYMBOL (If applicable) EC	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School		
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000			7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		
8a NAME OF FUNDING/SPONSORING ORGANIZATION		8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO	PROJECT NO	TASK NO
					WORK UNIT ACCESSION NO
11 TITLE (Include Security Classification) EFFICIENT TECHNIQUE FOR CALCULATING NEAR-FIELDS DUE TO 2-D SOURCES					
12 PERSONAL AUTHOR(S) ROSTANT, Richard A.					
13a TYPE OF REPORT Master's Thesis		13b TIME COVERED FROM _____ TO _____		14 DATE OF REPORT (Year, Month, Day) December 1990	
15 PAGE COUNT 137					
16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.					
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	electromagnetic scattering; near-fields		
19 ABSTRACT (Continue on reverse if necessary and identify by block number) Numerous methods exist to calculate near-fields from two-dimensional objects, however, relatively long computation times are generally required for reasonable accuracy. Computation is slowed primarily due to the calculation of near-fields using a singular kernel. The proposed work will develop an alternate, more efficient algorithm for calculating the near-fields from surface distributions. The Singularity Extraction Technique (SET) analytically extracts the contribution due to the near-singularity and implements the remaining portion numerically. Additionally, field contributions due to regions far removed from the field point are extracted out to further reduce the computational time. The implications here are a significant reduction in CPU time as well as improved accuracy. Computer programs are developed to implement and validate the SET. Testing includes comparison of the SET with analytic solutions to electromagnetic scattering for typical objects.					
20 DISTRIBUTION AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL JANASWAMY, Ramakrishna			22b TELEPHONE (Include Area Code) 408-646-3217		22c OFFICE SYMBOL EC/Js

DD Form 1473, JUN 86

Previous editions are obsolete

S/N 0102-LF-014-6603

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Approved for public release; distribution is unlimited.

**Efficient Technique for Calculating
Near-Fields Due to 2-D Sources**

by

Richard A. Rostant
Lieutenant, United States Naval Reserve
B.S., University of Oklahoma 1979

Submitted in partial fulfillment
of the requirements for the degree of

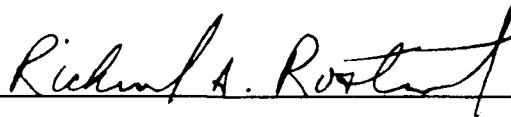
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

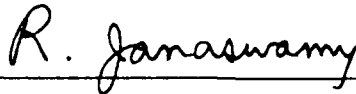
December 1990

Author:

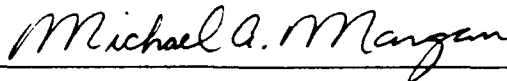


Richard A. Rostant

Approved by:



Ramakrishna Janaswamy, Thesis Advisor



Michael A. Morgan, Second Reader



Michael A. Morgan, Chairman

Department of Electrical and Computer Engineering

ABSTRACT

Numerous methods exist to calculate near-fields from two-dimensional objects, however, relatively long computation times are generally required for reasonable accuracy. Computation is slowed primarily due to the calculation of near-fields using a singular kernel. The proposed work will develop an alternate, more efficient algorithm for calculating the near-fields from surface distributions. The Singularity Extraction Technique (SET) analytically extracts the contribution due to the near-singularity and implements the remaining portion numerically. Additionally, field contributions due to regions far removed from the field point are extracted out to further reduce the computational time. The implications here are a significant reduction in CPU time as well as improved accuracy. Computer programs are developed to implement and validate the SET. Testing includes comparison of the SET with analytic solutions to electromagnetic scattering for typical objects.

Accession For	
NTIS - GRAM	<input checked="" type="checkbox"/>
PHC - TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Dist Branch	
Availability Codes	
Dist	
A-1	

TABLE OF CONTENTS

I. INTRODUCTION	1
A. BACKGROUND	1
B. PROBLEM	2
II. FORMULATION	4
A. NOMENCLATURE	4
B. GREEN'S FUNCTIONS	6
C. GREEN'S FUNCTION CONTOUR INTEGRAL	7
D. ASSOCIATED INEFFICIENCIES	9
E. SINGULARITY EXTRACTION TECHNIQUE	10
III. COMPUTER CODE DEVELOPMENT	17
A. IMPLEMENTATION	17
B. CIRCULAR CYLINDRICAL GEOMETRY	24
C. NEAR-FIELD PROGRAM	25
1. Program NEARFLD	25
2. Subroutine CIRCLE	26

3.	Subroutine SCAT	26
4.	Subroutine DSCAT	26
5.	Subroutine INCID	26
6.	Subroutine DINCID	27
7.	Subroutine ENDNODES	27
8.	Subroutine NODEPSI	27
9.	Subroutine REORD	27
10.	Subroutine CREORD	28
11.	Subroutine BES	28
D.	SINGULARITY EXTRACTION PROGRAM	28
1.	Subroutine SET	28
2.	Function CADRE (SIMP, TRAP)	29
3.	Functions ARGxx	29
4.	Function BESSJ0	30
5.	Function BESSY0	30
6.	Function BESSJ1	30
7.	Function BESSY1	30
E.	INPUT/OUTPUT	30
IV.	PARAMETER CHARACTERISTICS	32
A.	PHYSICAL CONSIDERATIONS	32
1.	Relative Permittivity (ϵ_r) and Permeability (μ_r)	32

2. Wavelength	33
3. Dimensions	33
B. NUMERICAL CONSIDERATIONS	34
V. TESTING AND VALIDATION	36
A. HARDWARE AND SOFTWARE	38
B. HANKEL FUNCTION APPROXIMATION	38
C. INCIDENT FIELD INTEGRATION	39
D. NEAR-FIELD CALCULATIONS	47
E. TIMED EVALUATIONS	77
VI. CONCLUSIONS	81
A. RESULTS	81
B. RECOMMENDATIONS AND EXTENSIONS	82
APPENDIX A. COORDINATE GENERATION ROUTINES	83
A. PROGRAM DESCRIPTION	83
B. PROGRAM LISTINGS	83
1. Program CIRCLE	83
2. Program SQUARE	83
3. Program SHELL	84
4. Program SLAB	85

APPENDIX B. INFINITE SERIES FIELD SOLUTIONS	87
APPENDIX C. NEARFLD PROGRAM	89
A. PROGRAM DESCRIPTION	89
B. PROGRAM LISTING	89
APPENDIX D. SINGULARITY EXTRACTION PROGRAM	104
A. PROGRAM DESCRIPTION	104
B. PROGRAM LISTING	104
APPENDIX E. EXPANDED FORM OF SET INTEGRAL TERM	116
APPENDIX F. INCIDENT FIELD INTEGRATION PROGRAM	118
A. PROGRAM DESCRIPTION	118
B. PROGRAM LISTING	118
LIST OF REFERENCES	122
INITIAL DISTRIBUTION LIST	123

LIST OF TABLES

TABLE 1. INCIDENT FIELD INTEGRATION PARAMETERS	36
TABLE 2. SCATTERED FIELD INTEGRATION PARAMETERS	37
TABLE 3. HANKEL FUNCTION APPROXIMATION	39
TABLE 4. AVERAGE SCATTERED FIELD	42
TABLE 5. ACCURACY OF SET	77
TABLE 6. SET ELAPSED TIME	78

LIST OF FIGURES

Figure 1. Two-Dimensional Cylindrical Object	5
Figure 2. Arbitrary Object	8
Figure 3. Infinitely Long Two-Dimensional Object	11
Figure 4. Linear Approximation of Contour $C2$	13
Figure 5. Discrete Version of Scattering Object	18
Figure 6. Contours $C1_k$ and $C2_k$ for $k = 1, 2$, and 3	21
Figure 7. Definition of Contour $C2_k$	22
Figure 8. Perimeter Contour of Hypothetical Object	40
Figure 9. Near-Field for Circular Cylinder, Incident Field Integration	43
Figure 10. Near-Field for Circular Cylinder, Incident Field Integration	44
Figure 11. Near-Field for Circular Cylinder, Incident Field Integration	45
Figure 12. Near-Field for Circular Cylinder, Incident Field Integration	46
Figure 13. Near-Field for Circular Cylinder, Incident Field Integration	48
Figure 14. Near-Field for Circular Cylinder, Incident Field Integration	49
Figure 15. Near-Field for Circular Cylinder, Incident Field Integration	50
Figure 16. Near-Field for Circular Cylinder, Scattered Field Integration	52
Figure 17. Near-Field for Circular Cylinder, Scattered Field Integration	53
Figure 18. Near-Field for Circular Cylinder, Scattered Field Integration	54
Figure 19. Near-Field for Circular Cylinder, Scattered Field Integration	55
Figure 20. Near-Field for Circular Cylinder, Scattered Field Integration	56

Figure 21. Near-Field for Circular Cylinder, Scattered Field Integration	57
Figure 22. Near-Field for Circular Cylinder, Scattered Field Integration	58
Figure 23. Near-Field for Circular Cylinder, Scattered Field Integration	59
Figure 24. Near-Field for Circular Cylinder, Scattered Field Integration	60
Figure 25. Near-Field for Circular Cylinder, Scattered Field Integration	61
Figure 26. Near-Field for Circular Cylinder, Scattered Field Integration	62
Figure 27. Near-Field for Circular Cylinder, Scattered Field Integration	63
Figure 28. Near-Field for Circular Cylinder, Scattered Field Integration	65
Figure 29. Near-Field for Circular Cylinder, Scattered Field Integration	66
Figure 30. Near-Field for Circular Cylinder, Scattered Field Integration	67
Figure 31. Near-Field for Circular Cylinder, Scattered Field Integration	68
Figure 32. Near-Field for Circular Cylinder, Scattered Field Integration	69
Figure 33. Near-Field for Circular Cylinder, Scattered Field Integration	70
Figure 34. Near-Field for Circular Cylinder, Scattered Field Integration	71
Figure 35. Near-Field for Circular Cylinder, Total Field Integration	73
Figure 36. Near-Field for Circular Cylinder, Total Field Integration	74
Figure 37. Near-Field for Circular Cylinder, Total Field Integration	75
Figure 38. Near-Field for Circular Cylinder, Total Field Integration	76
Figure 39. Near-Field for Circular Cylinder, Asymptotic Contribution Neglected	79
Figure 40. Near-Field for Circular Cylinder, Asymptotic Contribution Neglected	80

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to Professor Ramakrishna Janaswamy for his invaluable guidance throughout the course of this work.

This thesis is dedicated to my wife, Elizabeth, whom I consider equal partner in this accomplishment. Her love, understanding, and encouragement will not soon be forgotten.

I. INTRODUCTION

A. BACKGROUND

Electromagnetic scattering by dielectric objects is of great interest and is the primary focus of the present work. Predicting radar scattering characteristics of an arbitrary object is of particular interest in many areas of research today. Two specific areas which will benefit are Radar Target Classification, which exploits the signature of the targets scattered field for identification and Computer Aided Design (CAD) of electromagnetic structures.

Generally, three approaches exist to determine scattering characteristics [Ref. 1]:

1. Theoretical calculation
2. Dynamic experimentation
3. Static experimentation

The most practical method is certainly that of theoretical calculation. Although the benefits of theoretical methods are obvious, it is critical that an accurate and robust model be developed as with physical models of static experimentation. The objective of this research is to produce an accurate theoretical model for computing the scattered fields very close to a scattering body given the surface fields.

B. PROBLEM

The requirement to understand and predict scattering characteristics, namely, the scattering width or radar cross section, of a two-dimensional (2-D) object given its physical parameters, is the overall goal of this work. This is accomplished by first determining near-fields of the object, which are directly responsible for the charges and currents induced on the surface [Ref. 1]. Rigorous solutions to scattering by dielectric objects are available, but are restricted to few simple geometries [Ref. 2]. Numerous techniques exist to determine approximate near-field solutions such as physical optics, differential equations, and integral equations, to name a few [Ref. 3]. At one time, general solutions to electromagnetic boundary value problems were considered too unreliable and inaccurate, except for asymptotic cases [Ref. 4]. The advent of digital computers however, has facilitated techniques by which many of these problems can be solved.

Quantities associated with the near-fields are sources, surface currents and surface charges [Ref. 3]. The fields of interest associated with the scattering body can be represented by integrals in terms of these quantities. Numerical solutions to these integrals describing near-fields from 2-D sources can be applied to arbitrary dielectric objects, however, evaluation of these integrals often proves difficult due to the presence of singular kernels in the integrands.

Alternate, more efficient forms of the integrals used to determine near-fields from 2-D sources will be developed. Singularities which occur as the source point approaches the field point are extracted analytically. Also, contributions to the near-

field along asymptotic regions of the object surface are subtracted out. Numerical algorithms of the resultant integrals are developed for arbitrary geometries. Testing and validation of the model is accomplished by comparison of results with those of exact theoretical solutions.

II. FORMULATION

As stated in the previous chapter, there is a need to efficiently evaluate the near-fields from 2-D cylindrical objects. Numerous methods exist for accomplishing this. One widely used approach is that of a Green's function contour integral, which is the approach taken here.

Direct numerical implementations of these integrals are possible through the use of digital computers, however they are generally inefficient due to 1) near-singular functions in the integrand, and 2) significant field contributions from the asymptotic regions of the contour (regions on the source, far away from the field point). An alternate approach to the Green's function integral is developed here. Since the integrand exhibits its singular behavior near the field point (designated by Q), an alternate expression is developed for this portion of the contour. Also, the contribution due to the asymptotic portion of the surface integral can be extracted analytically. These two manipulations of the Green's function integral should greatly increase the speed of the numerical integration with minimal affect on accuracy.

A. NOMENCLATURE

Consider the arbitrary 2-D cylindrical object of Figure 1. The shape of the object varies only in the x-y plane and is infinite in the z-direction. The perimeter of the object is defined by the contour C . It is required to calculate the

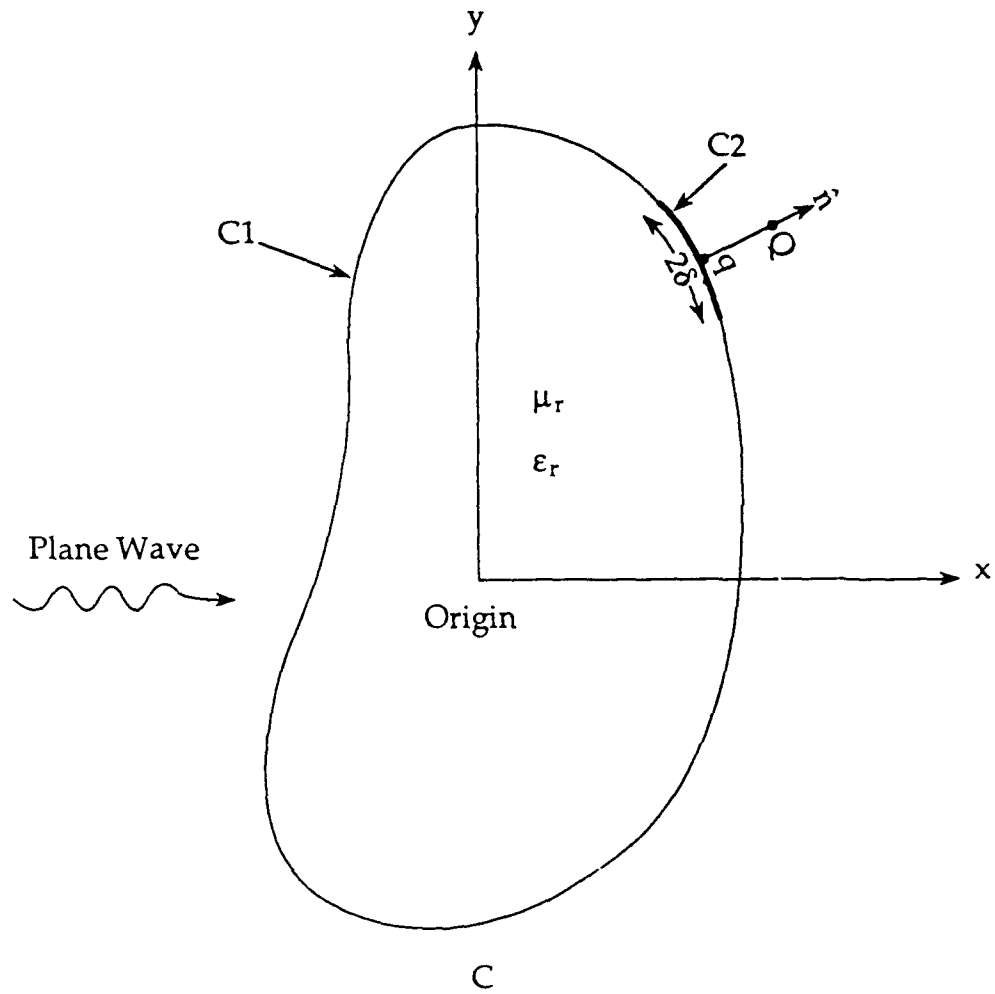


Figure 1. Two-Dimensional Cylindrical Object

field at point Q given the field and its normal derivative on C . In the subsequent development, the contour C is divided into two segments, $C1$ and $C2$. Segment $C2$ is a small portion of C which lies directly below the field point Q . Contour $C1$ is the remaining portion of C . Contour $C2$ is a distance of 2δ in arc length. The field point (Q) lies a distance d along the outward normal from the surface node point (q).

The incident wave is assumed to be a plane wave propagating in the direction of positive x-axis. The term *field* is defined to be E_z in the case of TM polarization and H_z in the case of TE polarization. The wavenumber in free space is denoted by k_0 , where $k_0 = \omega/c$, ω being the radian frequency of the incident wave, and c the velocity of the electromagnetic wave in free space. An $\exp(j\omega t)$ time dependence is assumed throughout. The total field, $\psi^{(t)}$, is written as the sum of the incident field, $\psi^{(i)}$, and the scattered field, $\psi^{(s)}$.

B. GREEN'S FUNCTIONS

Electromagnetic phenomena are concisely described by Maxwell's equations and appropriate boundary conditions [Ref. 3]. These equations can then be solved with a number of second-order uncoupled partial differential equations. The difficulty with this approach is that the solutions to these partial differential equations are, in general, slowly converging infinite series which yield little insight into the behavior of the specific function. An alternate and much more useful solution to the partial differential equations is obtained through the use of Green's functions which have proven invaluable in many areas of science and engineering. This approach provides

practical closed form solutions to differential equations, often in the form of integral equations.

The general concept of the Green's function technique is to obtain a solution to a partial differential equation by applying an impulse source function (Dirac delta) as a driving function [Ref. 3]. The response to this driving function is termed the Green's function. The solution to the differential equation is thus a superposition of the impulse response solution at each location, which in the limit is an integral. The Green's function is therefore analogous to the impulse response or transfer function of a linear system [Ref. 3]. It should be noted that the Green's function may occur in various forms, such as finite explicit functions or infinite series, depending upon the particular problem. All forms, however, yield the same results.

C. GREEN'S FUNCTION CONTOUR INTEGRAL

The scattered field, $\psi^{(s)}$, from an arbitrary object in free space, as in Figure 2, satisfying Helmholtz's equation [Ref. 5]

$$\nabla^2 \psi^{(s)} + k^2 \psi^{(s)} = 0 , \quad (1)$$

is

$$\psi^{(s)}(\bar{\rho}) = \oint \left[G(\bar{\rho}|\bar{\rho}') \frac{\partial \psi}{\partial n'} - \psi(\bar{\rho}') \frac{\partial G}{\partial n'} \right] dc' , \quad (2)$$

where ψ in the integrand may be either total or scattered field on the surface of the object, and $G(\bar{\rho}|\bar{\rho}')$ is the Green's function given by

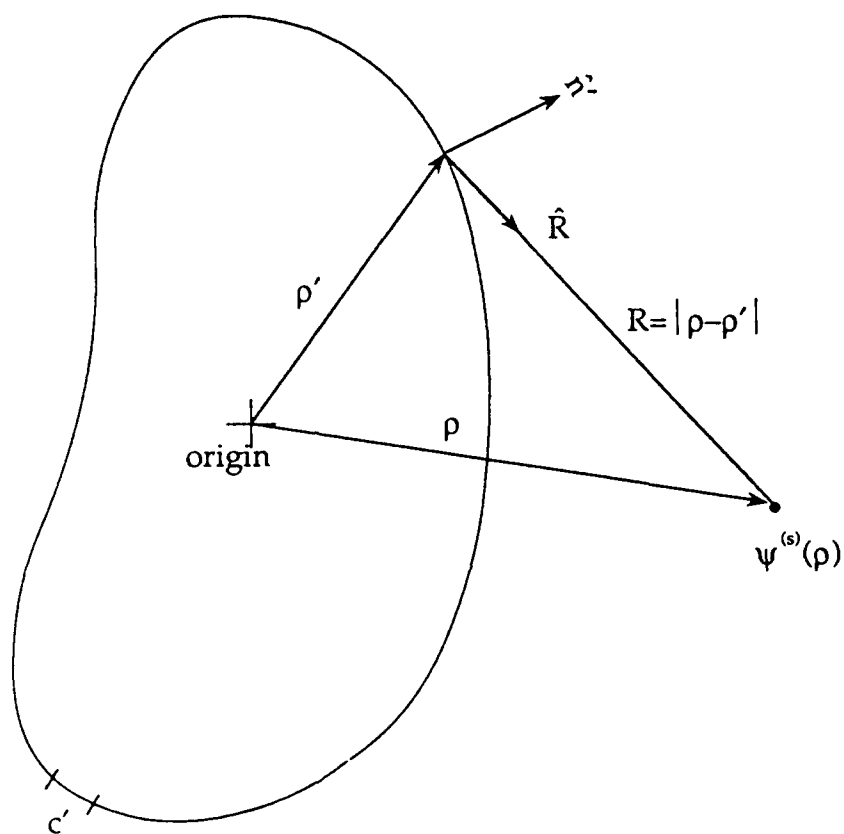


Figure 2. Arbitrary Object

$$G(\bar{\rho}|\bar{\rho}') = -\frac{1}{4j} H_0^{(2)}(k_0|\bar{\rho} - \bar{\rho}'|) , \quad (3)$$

and

$$\frac{\partial \Psi}{\partial n'} = \hat{n}' \cdot \nabla \Psi , \quad (4)$$

and

$$\frac{\partial G}{\partial n'} = \hat{n}' \cdot \hat{\rho} \frac{jk_0}{4} H_1^{(2)}(k_0|\bar{\rho} - \bar{\rho}'|) , \quad (5)$$

and $H_0^{(2)}$ and $H_1^{(2)}$ are Hankel functions of orders zero and one, respectively.

D. ASSOCIATED INEFFICIENCIES

Inherent difficulties exist in evaluating Equation (2) directly by means of numerical integration. The imaginary portion of the Hankel function rapidly approaches negative infinity as the argument approaches zero. This will be the case when the field point (Q) approaches the perimeter contour of the object and consequently, large CPU resources are required to compute the near-field surface integrations [Ref. 6]. This is primarily due to the large number of complex operations required for each step in the numeric quadrature.

In this thesis, an efficient scheme to compute the near-fields is developed. The general approach to this problem is to divide the object into two surface contours, $C1$ and $C2$, as in Figure 1 [Ref 7]. Contour $C1$ is numerically integrated without difficulty since R never approaches zero along this contour. An alternate,

more efficient method of calculating the field contribution due to contour $C2$ must then be derived. This is the primary emphasis of this work and is detailed in the next section.

The additional problem of large CPU requirements is addressed as well. Morgan [Ref. 6] proposes "to adaptively neglect the integration contributions outside a local neighborhood of the field point." Since the field contribution dies away with increasing distance from the field point, the integrations may be confined to a limited contour with minimal reduction in accuracy. This concept is addressed further in the development of the computer algorithm in Chapter III.

E. SINGULARITY EXTRACTION TECHNIQUE

Consider the infinitely long, two-dimensional arbitrary object of Figure 3. As previously stated, the scattered field at any point (Q) can be found from Equation (2) by integrating along the entire contour C . This contour can be divided into two distinct contours, $C1$ and $C2$. Equation (2) can be separated into two equations as

$$\psi^{(s)}(Q) = \int_{C1} \left(G \frac{\partial \psi}{\partial n'} - \psi \frac{\partial G}{\partial n'} \right) dl + \int_{C2} \left(G \frac{\partial \psi}{\partial n'} - \psi \frac{\partial G}{\partial n'} \right) dl . \quad (6)$$

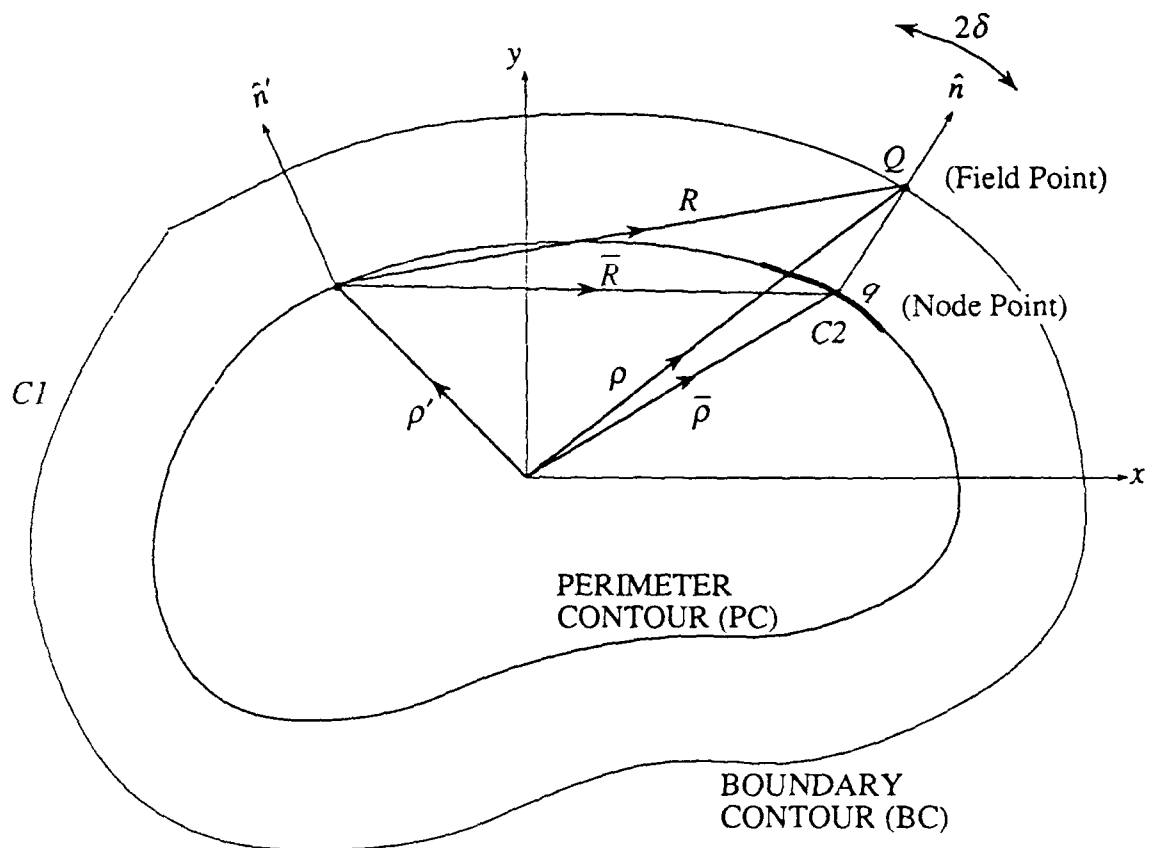


Figure 3. Infinitely Long Two-Dimensional Object

Numerical integration of the second term of Equation (6),

$$\int_{C_2} \left(G \frac{\partial \Psi}{\partial n'} - \Psi \frac{\partial G}{\partial n'} \right) dl = I_1 - I_2 , \quad (7)$$

where

$$I_1 = \int_{C_2} \left(G \frac{\partial \Psi}{\partial n'} \right) dl , \quad (8)$$

and

$$I_2 = \int_{C_2} \left(\Psi \frac{\partial G}{\partial n'} \right) dl , \quad (9)$$

is inefficient for near-field calculations, thus, an alternate form is desired [Ref. 7].

For small δ , contour C_2 approximates a linear segment as depicted in Figure

4. Using the small argument approximation of the Green's function [Ref. 5],

$$G(k_0 R) \sim -\frac{1}{4j} \left[1 - \frac{2j}{\pi} \ln(k_0 R) \right]_{k_0 R \rightarrow 0} , \quad (10)$$

Equation (8) can be written as

$$\int_{C_2} \left(G \frac{\partial \Psi}{\partial n} \right) dl \sim \frac{j}{4} \frac{\partial \Psi(q)}{\partial n} \int_{-\delta}^{\delta} \left[1 - \frac{2j}{\pi} \ln(k_0 \sqrt{d^2 + t^2}) \right] dt . \quad (11)$$

This leads to the final result,

$$\int_{C_2} \left(G \frac{\partial \Psi}{\partial n} \right) dl = -\delta \frac{\partial \Psi(q)}{\partial n} \left\{ \frac{1}{\pi} \left[1 - \frac{d}{\delta} \arctan\left(\frac{\delta}{d}\right) \right] - \frac{1}{2\pi} \ln[(k_0 \delta)^2 + (k_0 d)^2] - \frac{j}{2} \right\} . \quad (12)$$

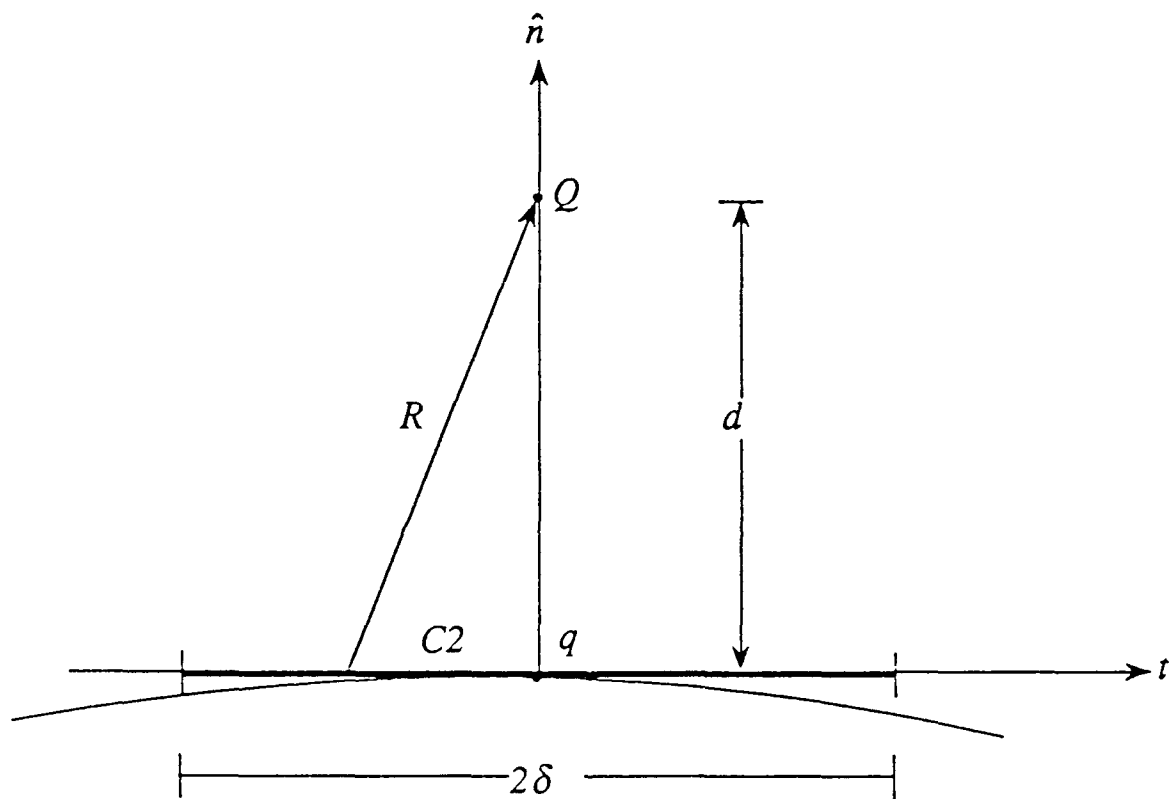


Figure 4. Linear Approximation of Contour $C2$

The small argument approximation for the normal derivative of the Green's function is [Ref. 5]

$$\frac{\partial G(k_0 R)}{\partial n'} \sim \frac{d}{2\pi R^2} \quad (13)$$

Thus, using Equation (13), it can be shown that Equation (9) can be written

$$\begin{aligned} \int_{C_2} \psi(q) \frac{\partial G}{\partial n'} &\sim \frac{d}{2\pi} \psi(q) \int_{-\delta}^{\delta} \frac{dt}{(t^2 + d^2)} \\ &- \frac{\psi(q)}{\pi} \arctan\left(\frac{\delta}{d}\right). \end{aligned} \quad (14)$$

Combining Equations (12) and (14) produces the desired alternate form of Equation (7),

$$\begin{aligned} \int_{C_2} \left(G \frac{\partial \psi}{\partial n'} - \psi \frac{\partial G}{\partial n'} \right) dl &= -\delta \frac{\partial \psi(q)}{\partial n} \left\{ \frac{1}{\pi} \left[1 - \frac{d}{\delta} \arctan\left(\frac{\delta}{d}\right) \right] \right. \\ &\quad \left. - \frac{1}{2\pi} \ln[(k_0 \delta)^2 + (k_0 d)^2] - \frac{j}{2} \right\} - \frac{\psi(q)}{\pi} \arctan\left(\frac{\delta}{d}\right). \end{aligned} \quad (15)$$

Substituting Equation (15) into Equation (6) yields,

$$\begin{aligned} \psi^{(s)}(Q) &= \int_{C_1} \left(G \frac{\partial \psi}{\partial n'} - \psi \frac{\partial G}{\partial n'} \right) dl - \delta \frac{\partial \psi(q)}{\partial n} \left\{ \frac{1}{\pi} \left[1 - \frac{d}{\delta} \arctan\left(\frac{\delta}{d}\right) \right] \right. \\ &\quad \left. - \frac{1}{2\pi} \ln[(k_0 \delta)^2 + (k_0 d)^2] - \frac{j}{2} \right\} - \frac{\psi(q)}{\pi} \arctan\left(\frac{\delta}{d}\right). \end{aligned} \quad (16)$$

At this point it should be noted that the integral in Equation (16) can be efficiently evaluated by means of numerical integration. The remaining terms represent the contribution from contour C_2 . The effects of the field point approaching the object

surface is represented by taking the limit of Equation (16) as d approaches zero, which yields the scattered field on the perimeter contour

$$\psi^{(s)}(q) = \int_{Cl} \left(\bar{G} \frac{\partial \psi}{\partial n'} - \psi \frac{\partial \bar{G}}{\partial n'} \right) dl = \frac{\psi(q)}{2} - \delta \frac{\partial \psi(q)}{\partial n} \left[\frac{1}{\pi} - \frac{1}{2\pi} \ln(k_0 \delta)^2 - \frac{j}{2} \right], \quad (17)$$

where

$$\bar{G} = G(R)|_{d \rightarrow 0}, \quad (18)$$

and

$$\frac{\partial \bar{G}}{\partial n'} = \frac{\partial G(R)}{\partial n'} \Big|_{d \rightarrow 0}. \quad (19)$$

Subtracting Equation (17) from Equation (16) and rearranging, it can be shown that the scattered field at node Q on the boundary contour of the object is

$$\begin{aligned} \psi^{(s)}(Q) = & \int_{Cl} \left[(G - \bar{G}) \frac{\partial \psi^{(s)}}{\partial n'} - \psi^{(s)} \frac{\partial (G - \bar{G})}{\partial n'} \right] dl \\ & - \psi^{(s)}(q) \left[\frac{1}{\pi} \arctan\left(\frac{\delta}{d}\right) - \frac{3}{2} \right] \\ & + \frac{\delta}{\pi} \frac{\partial \psi^{(s)}(q)}{\partial n} \left\{ \frac{d}{\delta} \arctan \frac{\delta}{d} + \frac{1}{2} \ln \left[1 + \left(\frac{d}{\delta} \right)^2 \right] \right\}. \end{aligned} \quad (20)$$

If ψ in the original integral, Equation (2), is chosen to be the total field on the perimeter, Equation (20) becomes

$$\begin{aligned}
\psi^{(s)}(Q) = & \int_{C1} \left[(G - \overline{G}) \frac{\partial \psi^{(s)}}{\partial n'} - \psi^{(s)} \frac{\partial (G - \overline{G})}{\partial n'} \right] dl \\
& + \psi^{(s)}(q) - \psi^{(s)}(q) \left[\frac{1}{\pi} \arctan\left(\frac{\delta}{d}\right) - \frac{1}{2} \right] \\
& + \frac{\delta}{\pi} \frac{\partial \psi^{(s)}(q)}{\partial n} \left\{ \frac{d}{\delta} \arctan\left(\frac{\delta}{d}\right) + \frac{1}{2} \ln \left[1 + \left(\frac{d}{\delta}\right)^2 \right] \right\}.
\end{aligned} \tag{21}$$

Equations 20 and 21 represent more desirable forms of Equation (6), exclusive of the unruly integral over contour $C2$. In this form, the field contribution from contour $C1$ is easily evaluated by numerical integration. The contribution from $C2$ is now in the form of a simple analytic formula, thus eliminating the previous difficulties of integrating a near-singular function. This form permits efficient computer evaluation of the Green's function contour integral without sacrificing speed and accuracy.

III. COMPUTER CODE DEVELOPMENT

The ultimate goal of this work is to develop an efficient method of evaluating the near-zone scattered fields from an arbitrary 2-D object. Now that the analytic formulation is complete, a method of computer evaluation is presented here. Algebraic manipulation of the integrand in Equation (20) yields a form of the integral which can be easily programmed for the large number of iterations required. The program to evaluate the scattered field is designed to handle any 2-D object whose geometry is specified discretely. Initial evaluation was accomplished utilizing a group of subroutines to generate the required input parameters for circular cylindrical geometry. The circular cylinder is chosen due to its simple geometry as well as the availability of exact solutions for comparison with calculated results.

A. IMPLEMENTATION

In order to evaluate Equation (20) by means of a digital computer, a discrete version of the scattering object is considered as seen in Figure 5. The object is initially divided into N equal length segments S_k , defined by $N + 1$ nodes on the perimeter contour C . The scattered field is found at each point Q_k on the boundary contour which is associated with a node point q_k on the perimeter contour.

The SET program determines the scattered field, $\psi^{(s)}(Q_k)$, for the k -th field point Q_k by summing the contributions due to contours $C1_k$ and $C2_k$. Contours $C1_k$

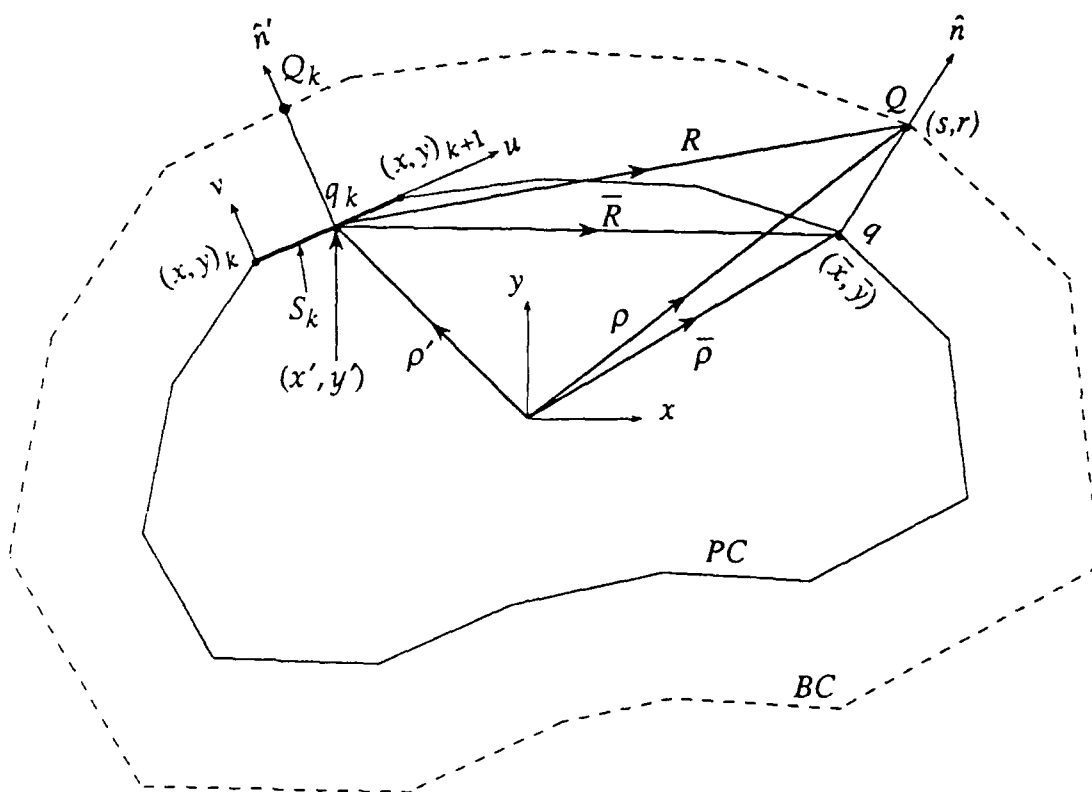


Figure 5. Discrete Version of Scattering Object

and $C2_k$ vary according to the specific point Q_k in question as illustrated in Figure 3. The contribution to the field, $\psi^{(s)}(Q_k)$, at Q_k due to contour $C2_k$ is determined by the analytic terms of Equation (20). Contribution from contour $C1_k$ is found by means of numerical integration over each segment, S_k , which make up the contour. The total field contribution due to $C1_k$ is the sum of the integrations. The resultant scattered field, $\psi^{(s)}(Q_k)$, is thus the sum of the contributions from $C1_k$ and $C2_k$.

Before Equation (20) can be evaluated by means of numerical techniques, each value required as input must be specified discretely. Each object considered must be described geometrically and electrically by discrete quantities. Both contours (i.e. perimeter and boundary) of the object are defined by a set of cartesian coordinates which are individually called nodes. Discrete field quantities at each node are determined as well.

The discrete geometry of the object must first be determined. Equally spaced coordinate nodes for typically shaped 2-D objects such as a circle, shell, square or slab can be determined using routines similar to those in Appendix A. The input consists of the number of nodes desired, the radius of the object, and the distance between the perimeter and boundary contours known as the offset distance. The output is the (x,y) coordinates of the perimeter contour and the (s,r) coordinates of the boundary contour. The coordinates for each node are stored in the $(N \times 4)$ matrix

$$\text{XYSR} = \begin{bmatrix} x_1 & y_1 & s_1 & r_1 \\ x_2 & y_2 & s_2 & r_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & s_N & r_N \end{bmatrix}. \quad (22)$$

For the initial development, the scattered field, $\psi^{(s)}$, and its normal derivative, $\partial\psi^{(s)}/\partial n$ on the perimeter contour are determined using infinite series methods outlined in Appendix B. The values of ψ and $\partial\psi/\partial n$ corresponding to each perimeter node point are determined and placed in the $(N \times 2)$ matrix

$$\text{PSI} = \begin{bmatrix} \psi_1 & \frac{\partial\psi_1}{\partial n} \\ \psi_2 & \frac{\partial\psi_2}{\partial n} \\ \vdots & \vdots \\ \psi_N & \frac{\partial\psi_N}{\partial n} \end{bmatrix}. \quad (23)$$

A set of end nodes for contours CI_k and $C2_k$, as in Figure 6, must be determined for each boundary contour node q_k . The end nodes are found by extending a distance δ along the local tangent on either side of q_k as in Figure 7. Integration along contour CI is performed in the clockwise direction, thus the end nodes must remain distinct. The end nodes are therefore placed in the $(N \times 4)$ matrix

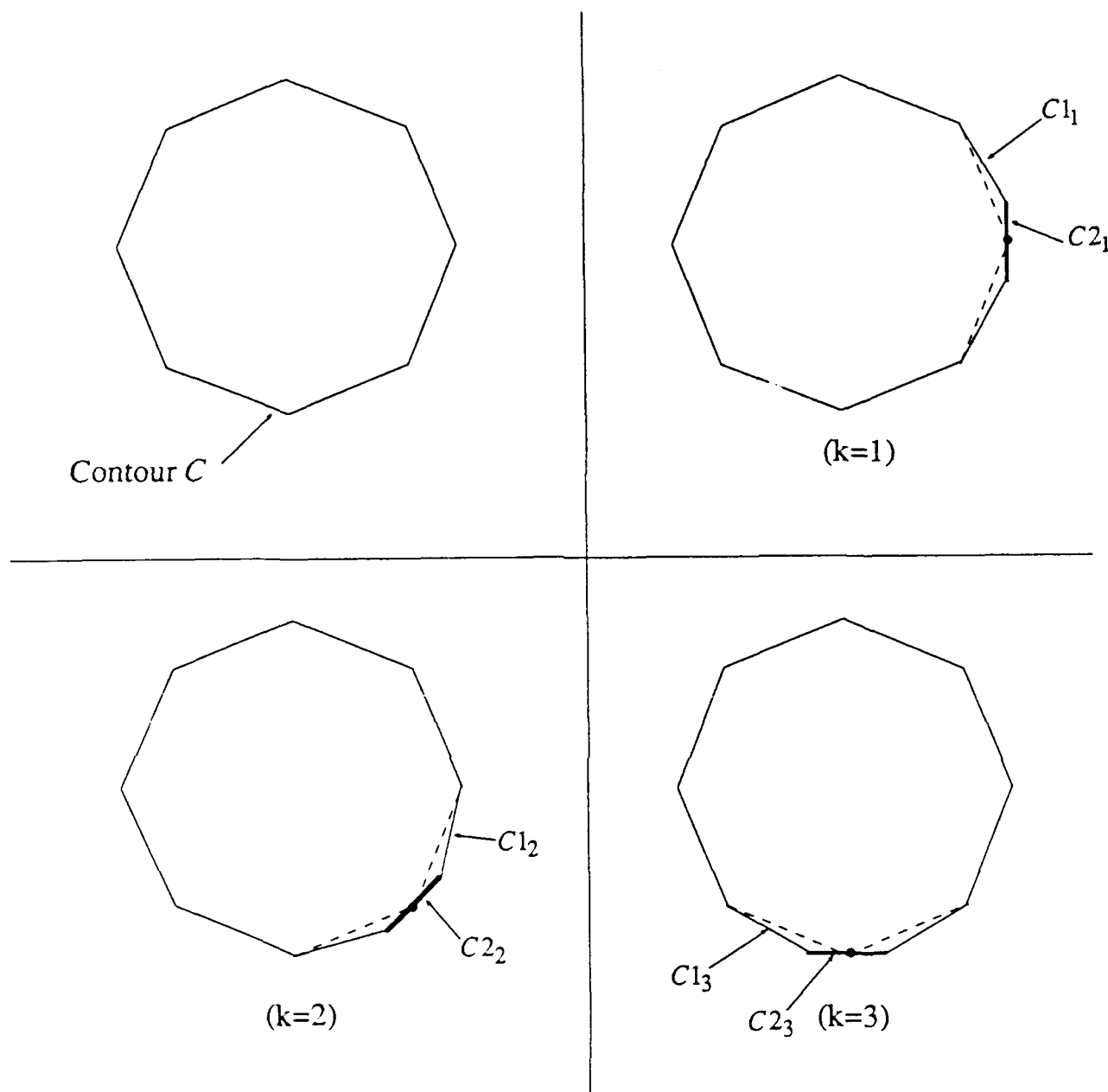


Figure 6. Contours $C1_k$ and $C2_k$ for $k = 1, 2$, and 3

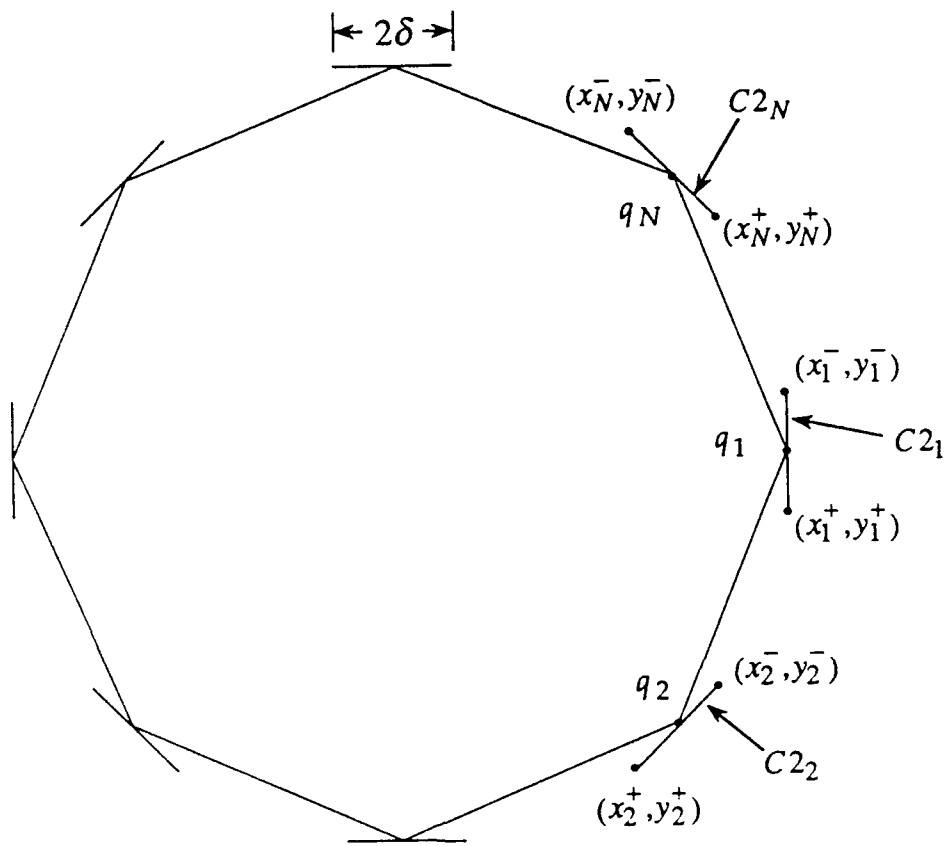


Figure 7. Definition of Contour $C2_k$

$$\text{PENDS} = \begin{bmatrix} x_1^+ & y_1^+ & x_1^- & y_1^- \\ x_2^+ & y_2^+ & x_2^- & y_2^- \\ \vdots & \vdots & \vdots & \vdots \\ x_N^+ & y_N^+ & x_N^- & y_N^- \end{bmatrix} \quad (24)$$

where (x_k^+, y_k^+) corresponds to the start node of contour CI and (x_k^-, y_k^-) correspond to the last node of CI , assuming a clockwise direction.

Corresponding values of ψ and $\partial\psi/\partial n$ are also required at each end node. These are obtained using a linear approximation and are stored in the $(N \times 4)$ matrix

$$\text{NEWPSI} = \begin{bmatrix} \psi(x_1^+, y_1^+) & \frac{\partial\psi(x_1^+, y_1^+)}{\partial n} & \psi(x_1^-, y_1^-) & \frac{\partial\psi(x_1^-, y_1^-)}{\partial n} \\ \psi(x_2^+, y_2^+) & \frac{\partial\psi(x_2^+, y_2^+)}{\partial n} & \psi(x_2^-, y_2^-) & \frac{\partial\psi(x_2^-, y_2^-)}{\partial n} \\ \vdots & \vdots & \vdots & \vdots \\ \psi(x_N^+, y_N^+) & \frac{\partial\psi(x_N^+, y_N^+)}{\partial n} & \psi(x_N^-, y_N^-) & \frac{\partial\psi(x_N^-, y_N^-)}{\partial n} \end{bmatrix} \quad (25)$$

At this point, the quantities required for integration on CI_k are available but must be properly arranged for each field point Q_k considered. A new $(N \times 2)$ matrix of nodes describing contour CI_k is defined as

$$\text{PNODC2} = \begin{bmatrix} x_k^+ & y_k^+ \\ x_{k+1} & y_{k+1} \\ \vdots & \vdots \\ x_{k-1} & y_{k-1} \\ x_k^- & y_k^- \end{bmatrix}. \quad (26)$$

Similarly, the $(N \times 2)$ matrix of field quantities corresponding to the nodes of contour CI_k is defined as

$$\text{PSIC2} = \begin{bmatrix} \psi(x_k^+, y_k^+) & \frac{\partial \psi(x_k^+, y_k^+)}{\partial n} \\ \psi(x_{k+1}, y_{k+1}) & \frac{\partial \psi(x_{k+1}, y_{k+1})}{\partial n} \\ \vdots & \vdots \\ \psi(x_{k-1}, y_{k-1}) & \frac{\partial \psi(x_{k-1}, y_{k-1})}{\partial n} \\ \psi(x_k^-, y_k^-) & \frac{\partial \psi(x_k^-, y_k^-)}{\partial n} \end{bmatrix}. \quad (27)$$

These two matrices are redefined for each integration of contour CI_k corresponding to the desired field $\psi^{(s)}(Q_k)$.

B. CIRCULAR CYLINDRICAL GEOMETRY

One of the requirements for evaluating the Green's function contour integral, Equation (2), and hence the integral in Equation (20), is the determination of the field, ψ , and its normal derivative, $\partial\psi/\partial n$, on the object surface. This is by no means trivial, even for the simplest objects. However, exact solutions for ψ and $\partial\psi/\partial n$ do

exist for circular cylindrical geometry (see Appendix B). These solutions are in the form of convergent infinite series and are relatively straight forward to calculate by means of a computer [Ref 3]. Also, the coordinates for equally spaced nodes along the perimeter of the circle are quite simple to calculate. These are the primary reasons the circular cylinder is utilized for the initial testing and evaluation phase.

C. NEAR-FIELD PROGRAM

The software written to evaluate the accuracy of Equation (20) consists of two parts. The first part takes care of reading the input parameters, calculating the potentials on the perimeter and boundary contours, and establishing the proper sequence in the data matrices input to the second portion of the program. This is accomplished utilizing a series of subroutines which perform each of the initial calculations and data manipulations.

1. Program NEARFLD

NEARFLD is the main controlling program coupled with a group of component subroutines. Each routine is called to perform a specific task required to generate the input to the SET subroutine. Once the input data is available, the SET subroutine is called N times to calculate the value of $\psi(Q_k)$ for each discrete field point on the boundary contour. NEARFLD, as it appears in Appendix C, is set up for the circular cylindrical geometry. It can easily be converted to handle any geometry by replacing CIRCLE with an alternate coordinate generation subroutine from Appendix A.

2. Subroutine CIRCLE

This subroutine computes the (x,y) coordinates of the discrete node points on the circular perimeter and boundary contours. The input parameters consist of the normalized radius of the perimeter contour, the number of discrete nodes, and the normalized offset distance between the perimeter and boundary contours. The output is a matrix containing the node coordinates on the respective contour.

3. Subroutine SCAT

This subroutine utilizes the method outlined in Appendix B to calculate an exact solution for the scattered fields from a dielectric circular cylinder. SCAT is initially called to calculate the fields on the boundary contour which are used for comparison with the fields calculated by the SET. It is again used to find the fields on the perimeter contour which are input to the SET.

4. Subroutine DSCAT

DSCAT calculates an exact solution of the normal derivative of the scattered field, $\partial\psi^{(s)}/\partial n$, on the surface of the circular cylindrical object utilizing the method of Appendix B. This value is required input to the SET.

5. Subroutine INCID

Similar to SCAT, subroutine INCID calculates the exact solutions for the incident field from a plane wave. This routine is only required when evaluating Equation (21), where the total field is used on the right side of the equation.

6. Subroutine DINCID

DINCID calculates the exact solution of the normal derivative of the incident field for a plane wave impinging on an object. It is utilized only when using Equation (21) to calculate scattered field.

7. Subroutine ENDNODES

For each point q_k , the endpoints of the contours $C1_k$ and $C2_k$ must be defined. The function of ENDNODES is to calculate the (x,y) coordinates of these endpoints. This is accomplished by calculating the (x,y) coordinates of the points $\pm \delta$ away from the node q_k , along the tangent line as in Figure 5. These values are used by REORD as the first and last values in the coordinate matrix input to SET.

8. Subroutine NODEPSI

Since a new set of nodes are created by ENDNODES, corresponding values of ψ and $\partial\psi/\partial n$ must be calculated for each new endpoint. NODEPSI does this by making a linear approximation of each new value. These values are used by CREORD as the first and last values in the potential matrix input to the SET.

9. Subroutine REORD

For each boundary point Q_k , new perimeter contours $C1_k$ and $C2_k$ must be defined. REORD accomplishes this by manipulation of the coordinate matrix generated by CIRCLE. Contour $C1_k$ is now defined by endpoints from ENDNODES and the reordered coordinates, excluding node q_k . The new arrangement of coordinates is utilized by the SET. This procedure is repeated for every node.

10. Subroutine CREORD

This subroutine performs operations similar to those of REORD. A rearranged matrix containing values of ψ and $\partial\psi/\partial r$ corresponding to the reordered coordinate matrix is generated for every node.

11. Subroutine BES

This subroutine calculates the ordinary Bessel functions $J_n(X)$ and $Y_n(X)$, and their first derivatives for integer order "n" from $n = 0$ to N for the real argument X [Ref. 8]. This subroutine is utilized by SCAT and DSCAT.

D. SINGULARITY EXTRACTION PROGRAM

The second part of the main program is the actual implementation of Equation (20). It consists of a group of subroutines and functions (Appendix D) which calculate the near-fields, $\psi^{(s)}(Q_k)$, for a lossless dielectric object, given the appropriate input data. This group of subroutines can easily be incorporated into any main program which requires the evaluation of a "near-field" Green's function contour integral. The subroutine which comprise this portion of the program are described below.

1. Subroutine SET

This subroutine is designed to solve the series of expressions listed in Appendix E which represent an expanded form of Equation (20). For each field point considered, the subroutine first calculates the analytic portion of Equation (20) which is the field contribution for contour $C2_k$. Next, the field contribution from each

segment of contour CI_k is calculated by solving each of the 12 integrals in Appendix E. (Note: A correction factor of $[-1]$ is required for the integral term of Equations (20) and (21). The cause of this abnormality was not determined at the time of this publication.) When the source point is greater than some EPS1 from the field point, the integrands in Equation (20) become quite small resulting in an insignificant field contribution from the individual segment. In this case, the integration is bypassed, thus reducing CPU time. The total field contribution from CI_k is the sum of the integration along each segment of the contour. The field contribution from CI_k and $C2_k$ are added yielding the scattered field, $\psi^{(s)}(Q_k)$.

2. Function CADRE (SIMP, TRAP)

Due to the discontinuous nature of many of the integrands in Appendix E, an adaptive integration scheme may be required. The adaptive numerical integration routine, CADRE [Ref. 9], is used here to successfully handle all jump discontinuities encountered. The integration routines SIMP and TRAP [Ref. 8], which apply Simpson's rule and the Trapezoid rule, respectively, can be used in the place of CADRE depending on the nature of the integrand. For most cases evaluated in this work, the subroutine TRAP provided accurate results.

3. Functions ARGxx

These functions evaluate the associated integrand for each of the integrals of Appendix E.

4. Function BESSJ0

This subroutine is used to calculate the zero-order Bessel function required in the ARG functions [Ref. 8].

5. Function BESSY0

This subroutine is used to calculate the zero-order Neumann function required in the ARG functions [Ref. 8].

6. Function BESSJ1

This subroutine is used to calculate the first-order Bessel function required in the ARG functions [Ref. 8].

7. Function BESSY1

This subroutine is used to calculate the first-order Neumann function required in the ARG functions [Ref. 8].

E. INPUT/OUTPUT

Execution of the NEARFLD program for circular cylindrical geometry requires a set of input parameters used to define the system. The input is via a screen prompt for each of the following variables:

1. (A) Radius of the cylinder in meters
2. (F0) Frequency of the incident plane wave in Hertz
3. (N) Number of nodes considered

4. (L) 2^{L-1} iterations of the trapezoid rule per segment S_k
Note: This input is not required when utilizing SIMP or CADRE integration routines.
5. (FAC) Factor used to calculate the upper limit of the summation in the 'exact' scattered field computations
Note: A value of 1.5 to 2.0 is generally sufficient for accurate results.
6. (ER) Relative permittivity of the object
Note: This input can be modified to allow for complex values.
7. (MR) Relative permeability of the object
Note: This input can be modified to allow for complex values.
8. (DELTA) Length of the segment δ in meters in Figure 3
9. (OFFSET) Offset distance (d) in meters as in Figure 3
10. (EPS1) Factor used to determine if integration of a specific segment of contour CI is to be bypassed
Note: This factor is used to increase the speed of the near-field calculations.

The output of the program is written to four data files, each of which is designated by the user. The following is a description of the information contained in the individual data files:

1. The scattered field at each field point on the boundary contour as calculated by the 'exact' solution
2. The incident field at each node point on the perimeter contour as calculated by the 'exact' solution
3. The scattered field at each field point on the boundary contour as calculated by the NEARFLD and SET programs
4. The first and second terms of Equation (20)

IV. PARAMETER CHARACTERISTICS

The near-fields from an object are a function of many different parameters. These parameters are defined by the specific geometry and composition of the object, the incident field impinging on the object, and the field point considered. Artificial parameters are created as well in the formulation of the numerical technique used to solve the problem. In this chapter, each of the parameters, real and artificial, which have some affect on the output, are considered. The expected influence on the system, as well as the limitations each impose on it are discussed.

A. PHYSICAL CONSIDERATIONS

Certain physical characteristics are inherent to the particular case considered. These parameters are strictly a function of the physical properties of the object and the type of waveform present.

1. Relative Permittivity (ϵ_r) and Permeability (μ_r)

The primary affect of ϵ_r and μ_r on the system, is that of altering the wavelength within the dielectric object. The wavenumber in the dielectric is defined by the relationship

$$k_r = \frac{2 \pi f}{c} \sqrt{\epsilon_r \mu_r} , \quad (28)$$

where f is the frequency of the incident wave. Variation of ϵ_r or μ_r has the combined affect of adjusting the dimensions of the object by a factor of $\sqrt{\epsilon_r \mu_r}$. which in turn alters the observed surface currents on the object. This requires some adjustment of the number of node points considered in order to achieve a suitable sampling rate.

2. Wavelength

The wavelength (λ) of the incident wave also has a direct affect on the electrical dimensions within the dielectric. Longer wavelengths have less variation over the object and thus, in general, produce less variation in the electric currents on the surface of the dielectric. Higher frequency electromagnetic waves with shorter wavelengths excite more variation in the surface currents. This has the same net effect on the system as ϵ_r and μ_r . Thus, the number of nodes must be adjusted to produce an acceptable sampling rate.

3. Dimensions

The physical dimensions of the object obviously have an affect on the near-fields. The circular cylinder is completely defined by its radius (a). The offset distance (d) of Figure 3 defines the boundary contour. Each dimension can be expressed in terms of wavelength to provide a means of normalization. Utilizing this wavelength normalization, the object is completely described by the quantity $k_0 a$.

B. NUMERICAL CONSIDERATIONS

As a result of the derivation of Equation (20), a restriction is placed on $k_0 R$ and $k_0 \bar{R}$, where R and \bar{R} are defined in Figure 3. This is a result of the approximation of the Hankel function used to calculate the field contribution from contour $C2$. The argument, $k_0 R$, must be $\ll 1$. This is not due to near-field considerations, but simply a result of the small argument approximation of the Hankel function. The effects due to the value of $k_0 R$ on the system are investigated in Chapter V.

The quantity EPS1 is an adjustable parameter introduced in the SET program. It provides a means to bypass integrations of segments on $C1$ which provide negligible contribution to the near-field. This feature can be disregarded by making EPS1 larger than the diameter of the object.

The sampling rate (i.e., the number of nodes per wavelength) must be taken into consideration to produce accurate integration results. The linear approximation of ψ and $\partial\psi/\partial n$ on the perimeter require a large number of segments to describe these quantities on the surface of the object. This is accomplished by specifying a sufficient number of nodes, thus reducing the differential interval. The quantity $k_0 a \sqrt{\epsilon_r \mu_r}$ represents the number of wavelengths in the dielectric around the perimeter. A minimum of four nodes per wavelength,

$$\frac{k_0 a \sqrt{\epsilon_r \mu_r}}{N} \leq \frac{1}{4}, \quad (29)$$

should be used to obtain an accurate representation of the field quantities on the object surface.

V. TESTING AND VALIDATION

The difficulty in evaluating the validity of Equation (20) is due to a deficiency of established near-field solution techniques. Solutions to specific problems [Ref. 10] do however exist and are the focus of the validation phase. A number of different testing methods are developed and utilized in order to thoroughly validate the Singularity Extraction Technique. A variety of TM cases are evaluated, each of which is characterized by a set of representative data outlined in Table 1 and Table 2. The effects of each of the parameters on the system are also analyzed.

TABLE 1. INCIDENT FIELD INTEGRATION PARAMETERS

CASE	FIGURE	$k_0 a$	$k_0 b$	$k_0 d$	$k_0 \rho$	ϵ_r	μ_r	NODES
IF-1	9	0.6283	0.0628	0.0628	0.6912	2	1	36
IF-2	10	0.6283	0.0628	0.0628	0.6912	2	1	72
IF-3	11	6.2832	0.0628	0.0628	6.3460	2	1	72
IF-4	12	6.2832	0.0628	0.0628	6.3460	2	1	144
IF-5	13	62.8319	0.0628	0.0628	62.8947	2	1	72
IF-6	14	62.8319	0.0628	0.0628	62.8947	2	1	180
IF-7	15	62.8319	0.0628	0.0628	62.8947	2	1	360

TABLE 2. SCATTERED FIELD INTEGRATION PARAMETERS

CASE	FIGURE	$k_g \rho$	$k_g \delta$	$k_g d$	$k_a \rho$	ϵ_r	μ_r	NODES
SF-1	16	6.2832	0.0314	0.0628	6.3460	2	1	36
SF-2	17	6.2832	0.0628	0.0628	6.3460	2	1	36
SF-3	18	6.2832	0.3142	0.0628	6.3460	2	1	36
SF-4	19	6.2832	0.6283	0.0628	6.3460	2	1	36
SF-5	20	6.2832	0.0314	0.3142	6.5973	2	1	36
SF-6	21	6.2832	0.0628	0.3142	6.5973	2	1	36
SF-7	22	6.2832	0.3142	0.3142	6.5973	2	1	36
SF-8	23	6.2832	0.6283	0.3142	6.5973	2	1	36
SF-9	24	6.2832	0.0314	0.0628	6.3460	2	1	72
SF-10	25	6.2832	0.0628	0.0628	6.3460	2	1	72
SF-11	26	6.2832	0.3142	0.0628	6.3460	2	1	72
SF-12	27	6.2832	0.6283	0.0628	6.3460	2	1	72
SF-13	28	62.8319	0.0628	0.0628	62.8947	2	1	90
SF-14	29	62.8319	0.0628	0.0628	62.8947	2	1	180
SF-15	30	62.8319	0.0628	0.0628	62.8947	2	1	360
SF-16	31	6.2832	0.0628	0.0628	6.3460	5	5	18
SF-17	32	6.2832	0.0628	0.0628	6.3460	5	5	36
SF-18	33	6.2832	0.0628	0.0628	6.3460	5	5	72
SF-19	34	6.2832	0.0628	0.0628	6.3460	5	5	180
SF-20	35	6.2832	0.0314	0.0628	6.3460	2	1	36
SF-21	36	6.2832	0.0628	0.0628	6.3460	2	1	36
SF-22	37	6.2832	0.0314	0.0628	6.3460	2	1	72
SF-23	38	6.2832	0.0628	0.0628	6.3460	2	1	72
SF-24	39	6.2832	0.0628	0.0628	6.3460	2	1	180
SF-25	40	62.8319	0.0628	0.0628	62.8947	2	1	360

A. HARDWARE AND SOFTWARE

All programs utilized in this thesis are written in Fortran 77 language. An NDP Fortran-386 compiler is used to compile, link, and execute the code. All testing is conducted on an 80386-based personal computer employing a Weitek coprocessor.

B. HANKEL FUNCTION APPROXIMATION

The small argument approximation is made for the Hankel functions utilized in the development of Equation (20). This requires that the argument, $k_0 R$, be $\ll 1$, thus placing a bound on the term δ , which defines $C2$, and on the offset distance, d , specifically

$$\left[k_0 R - k_0 \sqrt{\delta^2 + d^2} \right] \ll 1 . \quad (30)$$

The question which arises is, how close to zero must the argument be for acceptable accuracy of the Hankel function approximation. A comparison was made between the small argument approximation and a direct power series solution of the Hankel function $H_0^{(2)}(k_0 R)$. The results for several values of the argument are listed in Table 3. The relative error of the approximation is quite acceptable for arguments ($k_0 R$) of less than 0.3. In general, this restriction was adhered to for all testing and validation conducted within this research.

TABLE 3. HANKEL FUNCTION APPROXIMATION

$k_p R$	Relative Error
0.01	0.023
0.02	0.027
0.03	0.029
0.04	0.031
0.05	0.033
0.08	0.036
0.10	0.037
0.30	0.037
0.50	0.059
0.80	0.192
1.00	0.326

C. INCIDENT FIELD INTEGRATION

One way to test the performance of the SET is to compare its results with those of proven theory. Consider the case depicted in Figure 8, where the hypothetical boundary D is in a homogenous medium (constants ϵ_r and μ_r). Since there is no material interface, the scattered field due to D is zero and the only field present is the incident field. Next, consider determining the scattered field, $\psi^{(s)}(Q)$, using Equation (21). In this case, the total field on the right side of the equation is equal to the incident field alone. Evaluation of Equation (21) should yield $\psi^{(s)}(Q) = 0$.

The computer program INTEST (Appendix F) was developed to evaluate Equation (2) for $\psi^{(i)} = \psi^{(i)}$. The term 'exact', in the figures that follow, indicates the near-field calculation using Equation (2). Equation (21), which considers the total

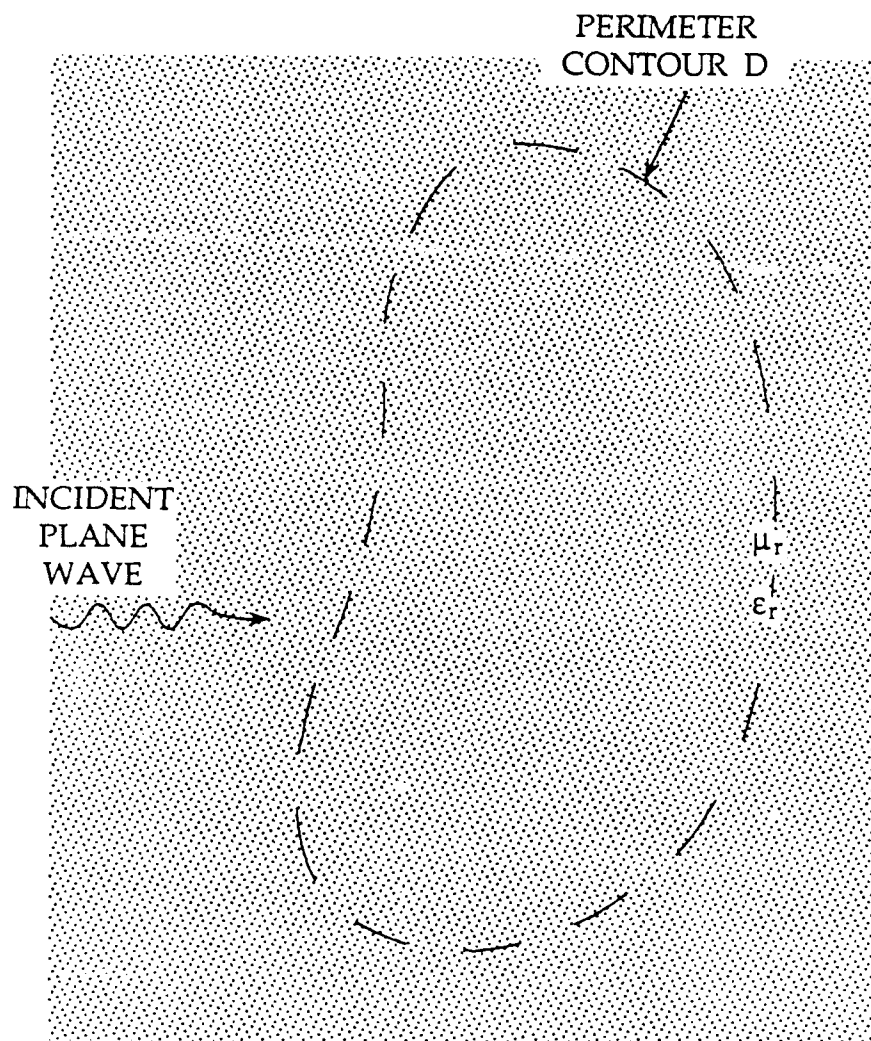


Figure 8. Perimeter Contour of Hypothetical Object

field ($\psi = \psi^{(i)}$) on the perimeter contour, was evaluated for the case of the total field on the object equal to the incident field alone. As described above, the scattered field on the boundary contour for both procedures must be zero.

Several cases were considered, first using the program INTST and then the program NEARFLD for circular cylinders. Comparisons of the average magnitudes of the scattered field, $\psi^{(s)}(Q_k)$, calculated using each method are outlined in Table 4 where

$$\psi^{(s)}_{avg} = \frac{1}{N} \sum_{k=1}^N |\psi^{(s)}(Q_k)|. \quad (31)$$

Note that the values for each are of the same order of magnitude in cases IF-1 - IF-4. The values also approach zero as the number of nodes is increased. This is due to the better approximation of ψ corresponding to the increased sampling rate as discussed in Chapter IV.

TABLE 4. AVERAGE SCATTERED FIELD

CASE	$\psi^{(s)}_{avg}$	
	Exact	SET
IF-1	~ 0	0.02786
IF-2	~ 0	0.02683
IF-3	0.02024	0.02887
IF-4	0.00092	0.02895
IF-5	22.94755	0.22323
IF-6	12.21451	0.04413
IF-7	0.01239	0.02950

Figure 9 depicts the low frequency ($f = 30$ MHz) results for a dielectric cylinder with $k_0 a = 0.628$. The scattered near-field on the boundary contour ($k_0 \rho = 0.691$) calculated by INTEST is equivalent to zero as expected. The scattered near-field calculated using the SET is shown as well. Comparison of the two methods for this near-field case exhibit good agreement with theoretical results, specifically, zero scattered field. Figure 10 contains the results for this case with an increased number of nodes. Both cases produce good results since an adequate number of sampling points were considered for each.

Figure 11 shows the near-field for the medium frequency ($f = 300$ MHz) case with $k_0 a = 6.283$. Both methods, INTEST and SET, are equivalent to zero. Figure 12 is the same case for an increase in nodes. Again, there is no significant divergence since, in both case, the sampling rate was sufficient to obtain an accurate solution.

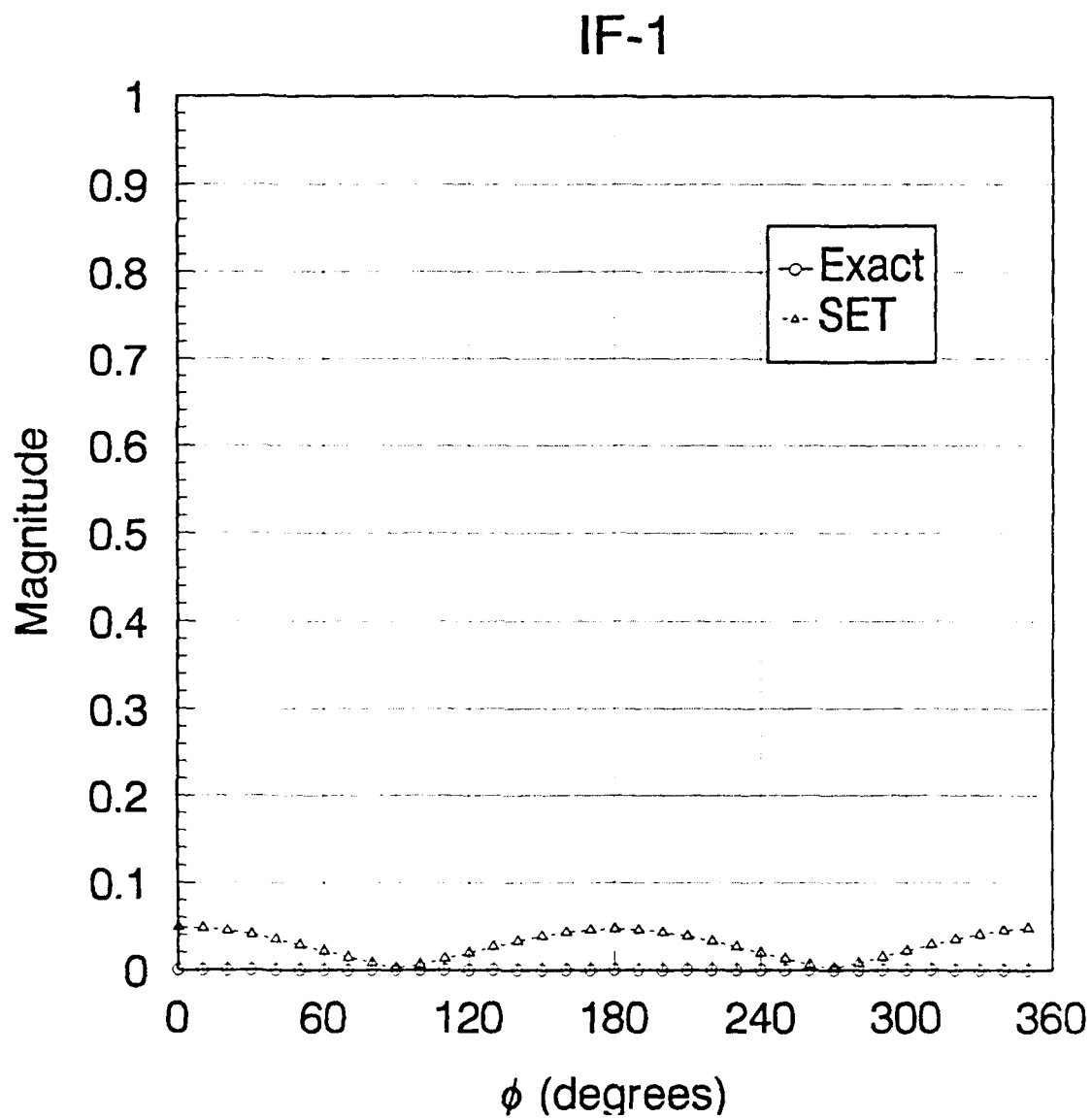


Figure 9. Near-Field for Circular Cylinder, Incident Field Integration

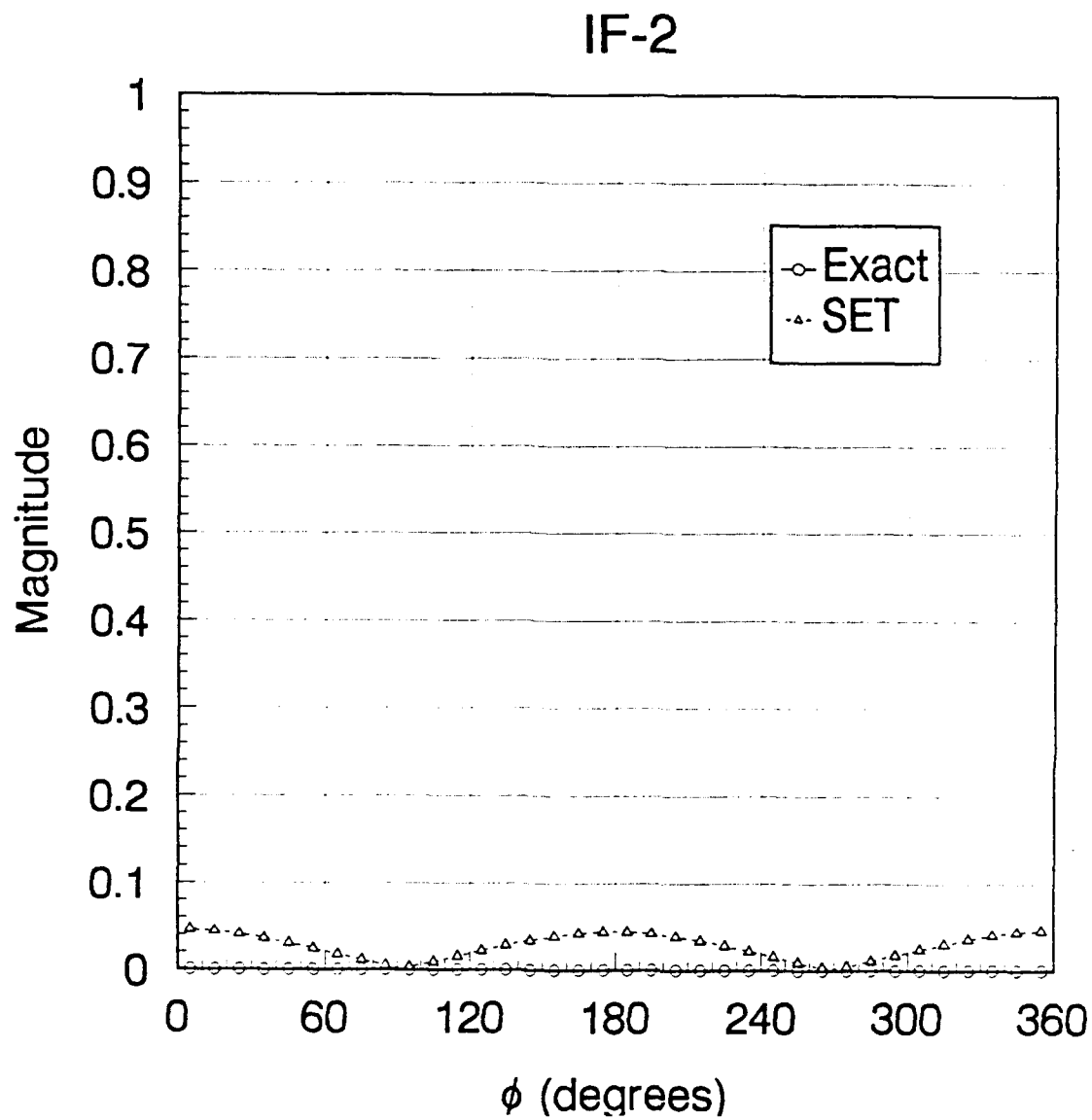


Figure 10. Near-Field for Circular Cylinder, Incident Field Integration

IF-3

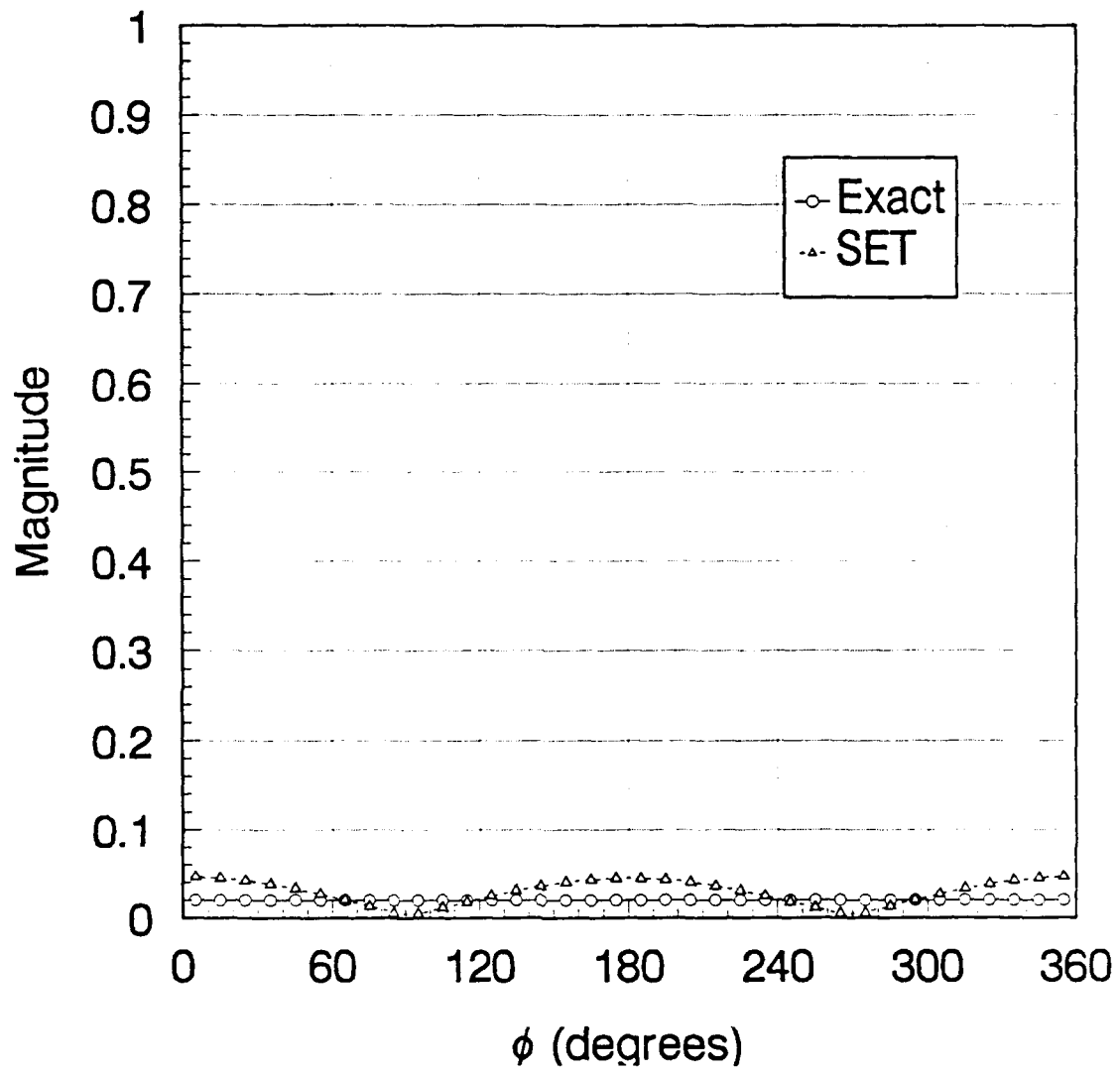


Figure 11. Near-Field for Circular Cylinder, Incident Field Integration

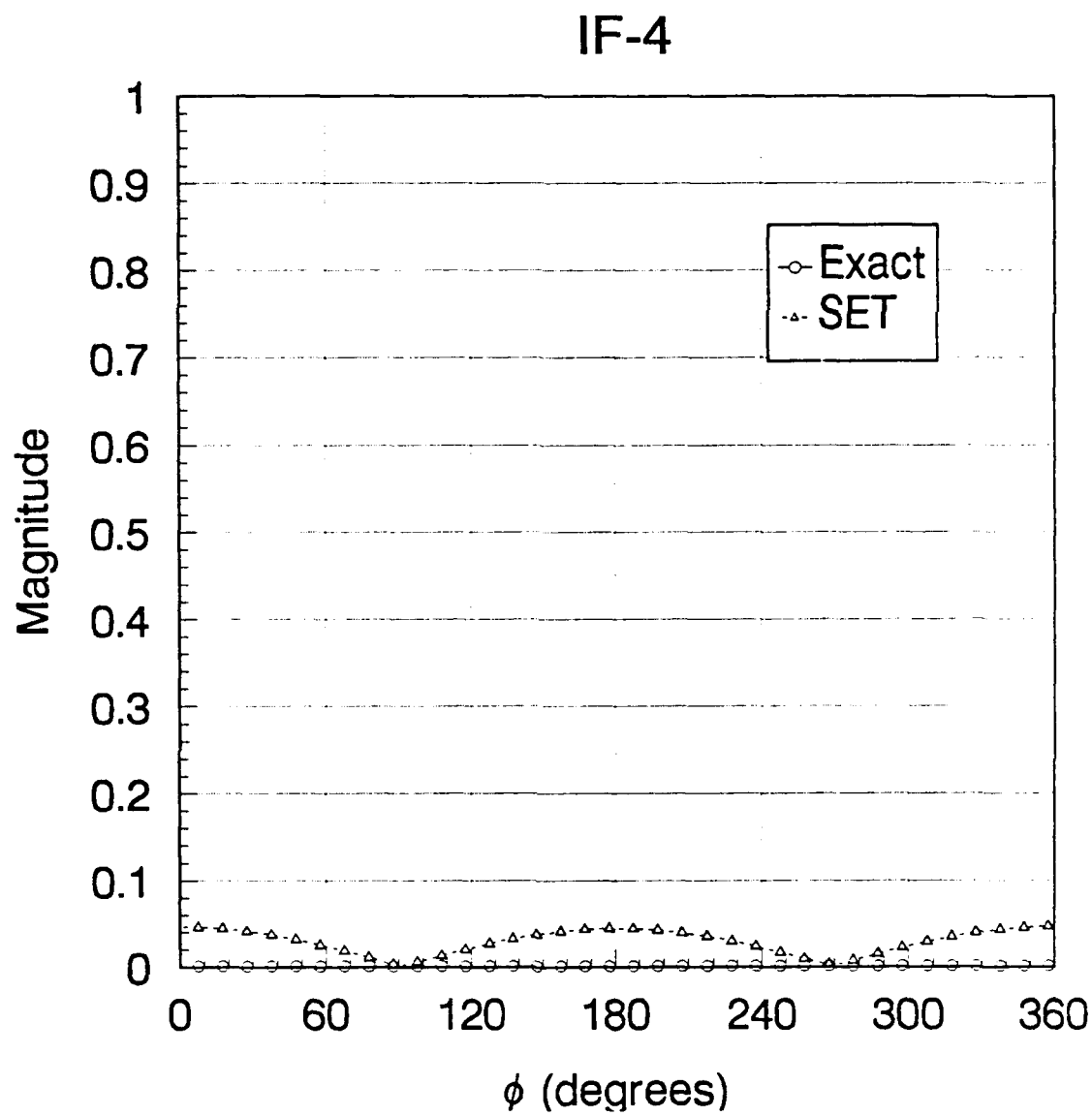


Figure 12. Near-Field for Circular Cylinder, Incident Field Integration

The high frequency ($f = 3$ GHz) cases for $k_0 a = 62.832$ appear in Figures 13-15. The results obtained from INTEST and SET for two undersampled cases appear in Figures 13 and 14. Both methods produce large inaccuracies due to undersampling. Figure 15 depicts a high sampling rate which produces the near zero results expected with the exception of the forward scattering direction, where the results diverge somewhat. The method using INTEST has a rapid convergence to zero as the sampling rate is increased, where the SET is near zero, but still invalid.

Variation of other parameters have no significant effect on the above test cases.

D. NEAR-FIELD CALCULATIONS

The next phase of testing includes comparison of near-field calculations using the SET program with those of exact series solutions. Numerous cases were considered to observe the effects each parameter has on the near-field results. Again, circular cylindrical geometry was utilized due to the availability of accurate near-field solutions. Plots depicting the normalized near-fields for each case are included. The analytic and integral portions of Equation (20) are also plotted in some select cases to show that significant contributions from both terms of the equation are present in the SET generated near-field.

The initial tests were conducted for a medium frequency ($f = 300$ MHz) case with $k_0 a = 6.2832$. The object is a relatively simple circular dielectric cylinder with $\epsilon_r = 2$ and $\mu_r = 1$. The effect that the length of contour C_2 has on the SET is investigated by varying δ . It is anticipated that the accuracy of the SET will be

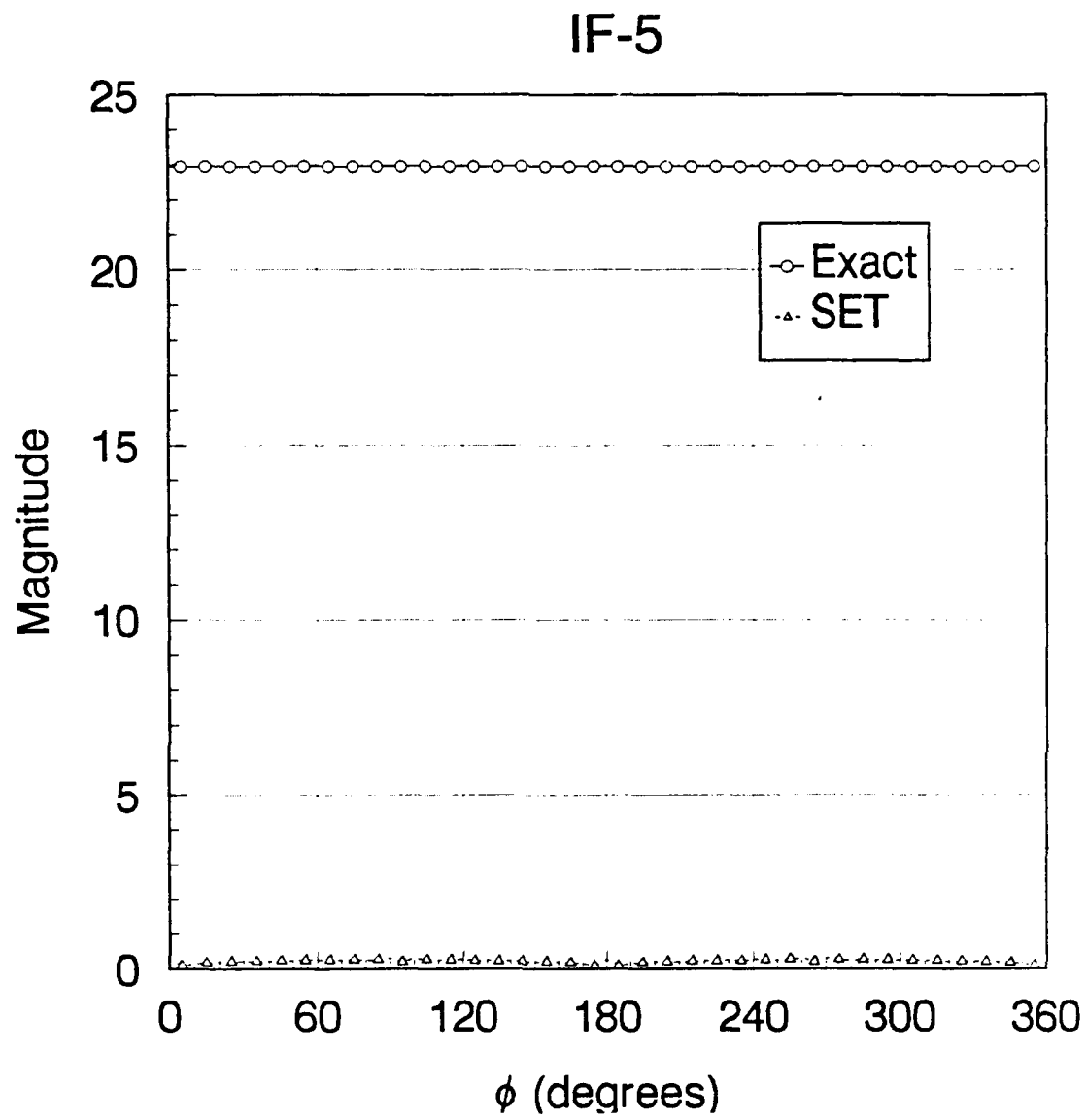


Figure 13. Near-Field for Circular Cylinder, Incident Field Integration

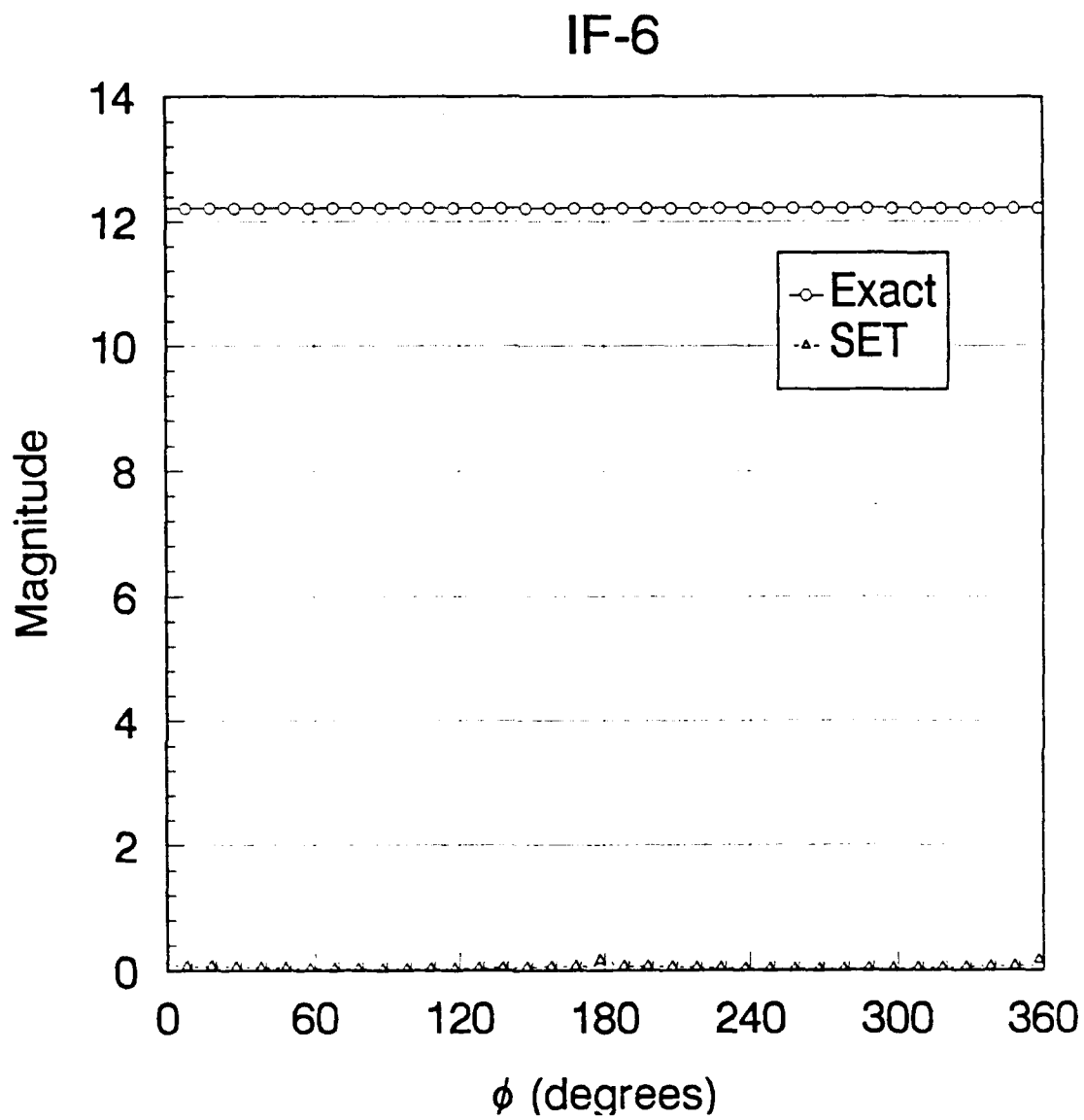


Figure 14. Near-Field for Circular Cylinder, Incident Field Integration

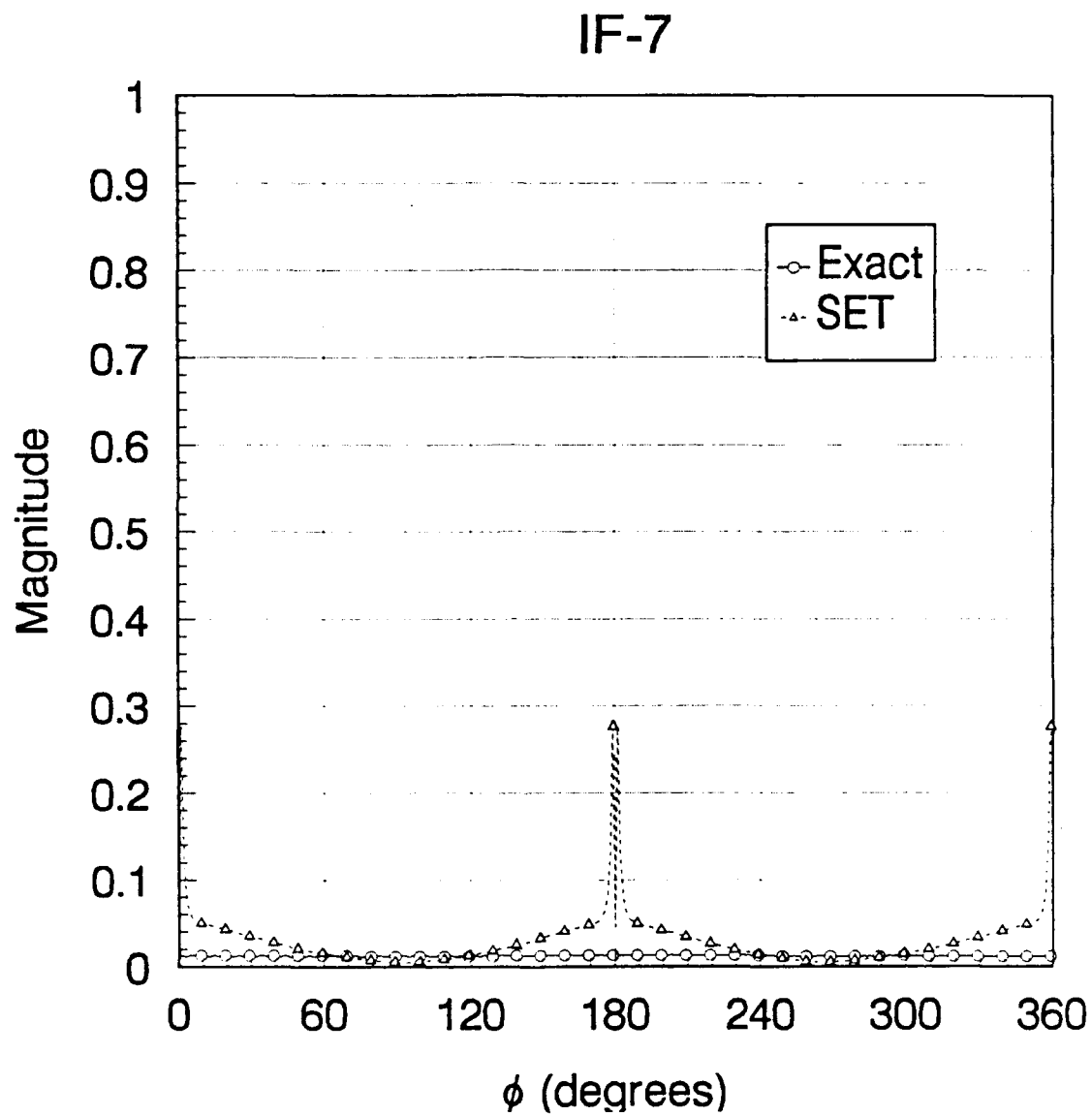


Figure 15. Near-Field for Circular Cylinder, Incident Field Integration

greater for smaller $C2$ since this is similar to integration over the entire contour C . Figure 16 is the case for $k_0\delta = 0.0314$, which corresponds to the smallest $C2$ considered. Notice the excellent agreement between the SET and exact near-field solutions. Figures 17-19 represent the near-field solutions as δ is increased. The results diverge slightly with increasing δ , but a strong correlation still exists between the SET and exact solutions. Next, the offset distance, d , was increased to $k_0d = 0.314$ for medium frequency ($f = 300$ MHz). Again, the contour distance parameter, $k_0\delta$, was varied between 0.0314 and 0.628. The results for each $k_0\delta$ considered appear in Figures 20-23. Generally, the near-fields calculated by the SET begin to diverge slightly from the exact solution. The solutions also become less accurate as $k_0\delta$ is increased. Obviously, increasing d has an affect on the accuracy of the SET which is due, in part, to the inequality $k_0R \ll 1$.

An increase in the number of nodes will provide a more accurate representation of the field quantities on the surface of the object. This corresponds to an increased sampling rate. It is anticipated that the SET program will produce a more accurate solution to the near-fields in this situation. Tests were conducted using parameters similar to those evaluated in Figures 15-18, with the exception of an increase in the number of nodes used. In each case, k_0d remains constant and $k_0\delta$ is varied.

Figure 24 shows the case for $k_0\delta = 0.0314$. As expected, the near-field calculated using the SET closely approximates the exact solution. The remaining three cases evaluated for increasing δ , shown in Figures 25-27, exhibit a slight divergence of the SET solution from the exact as δ is increased, but overall provides

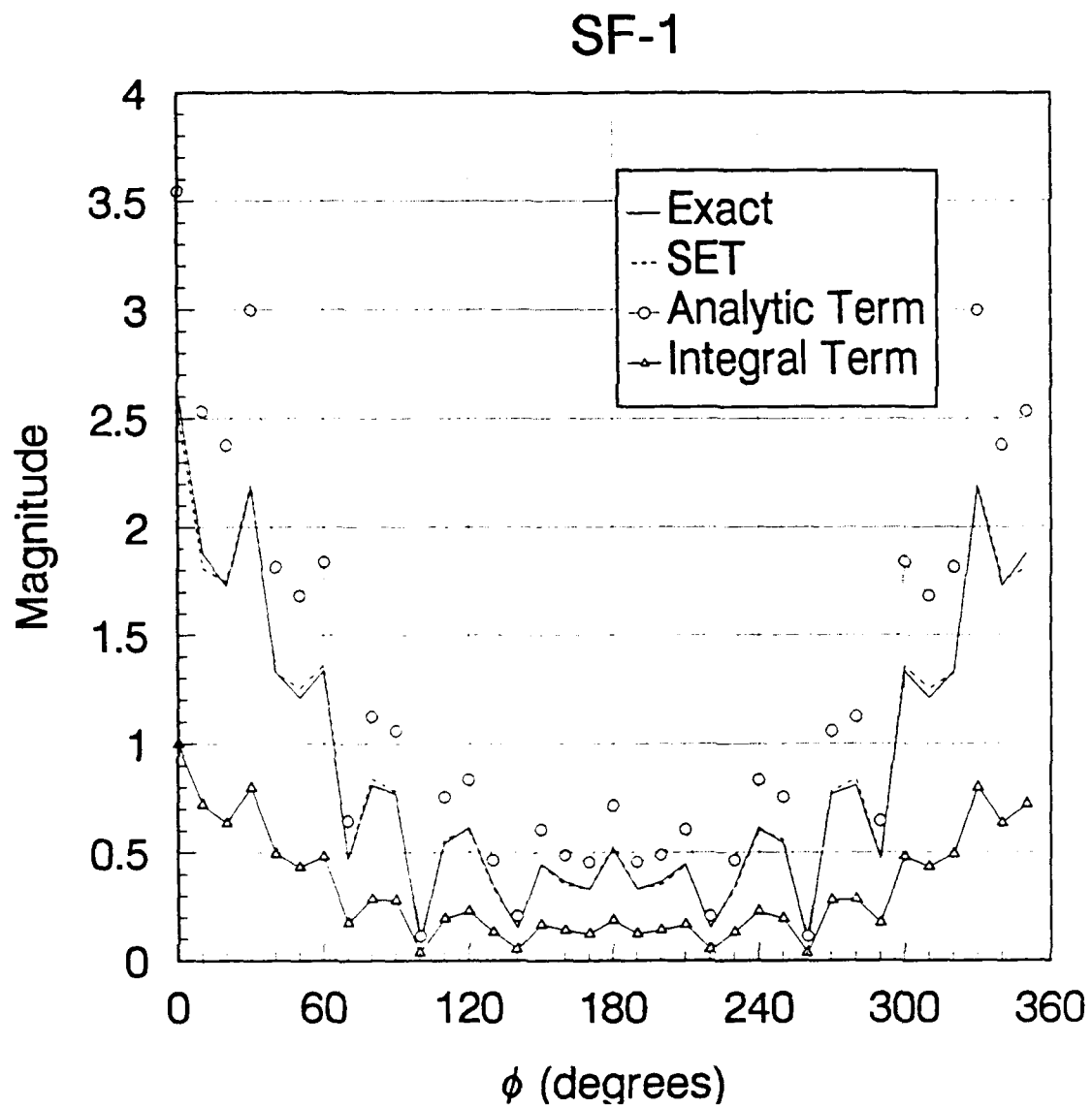


Figure 16. Near-Field for Circular Cylinder, Scattered Field Integration

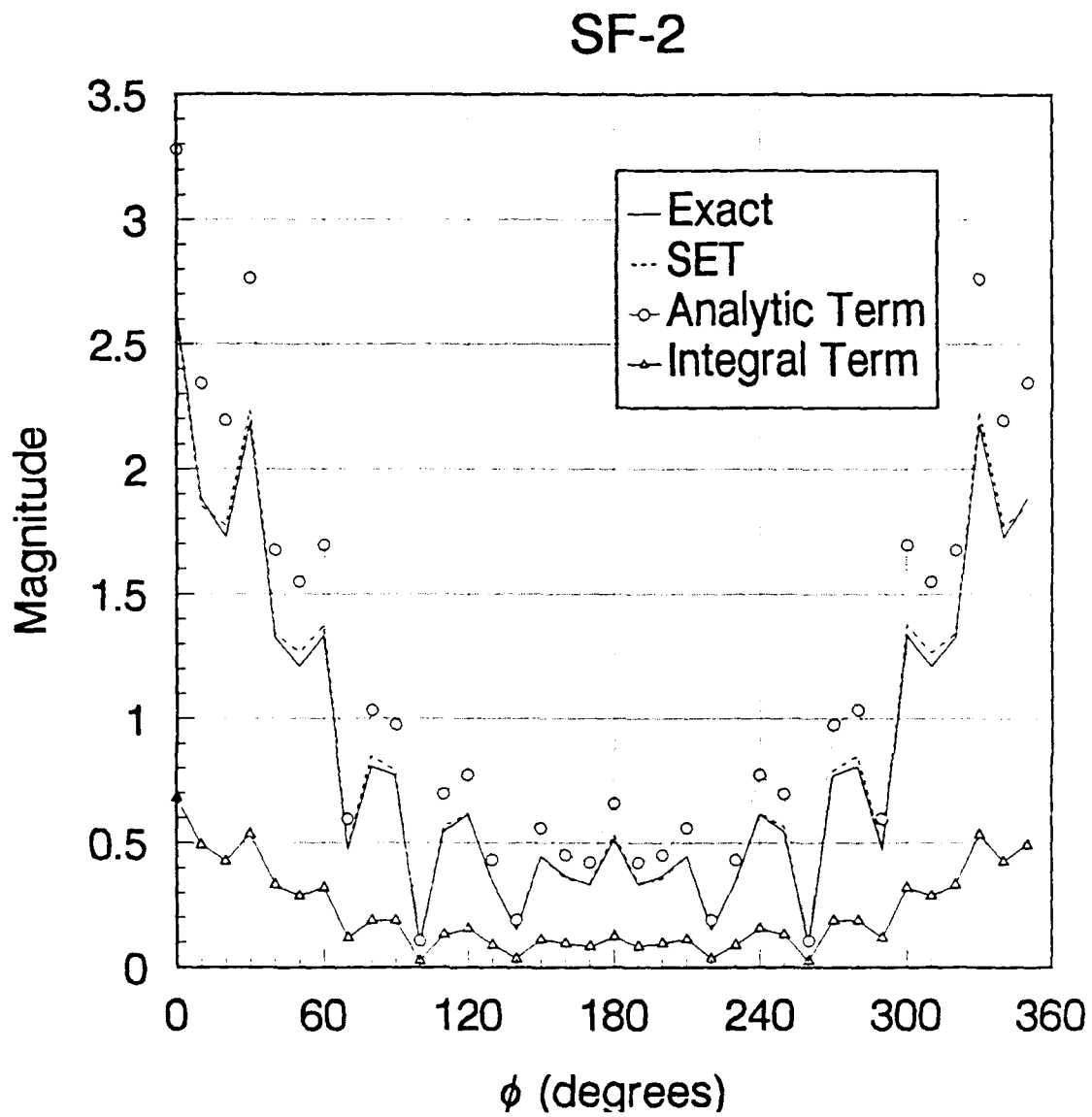


Figure 17. Near-Field for Circular Cylinder, Scattered Field Integration

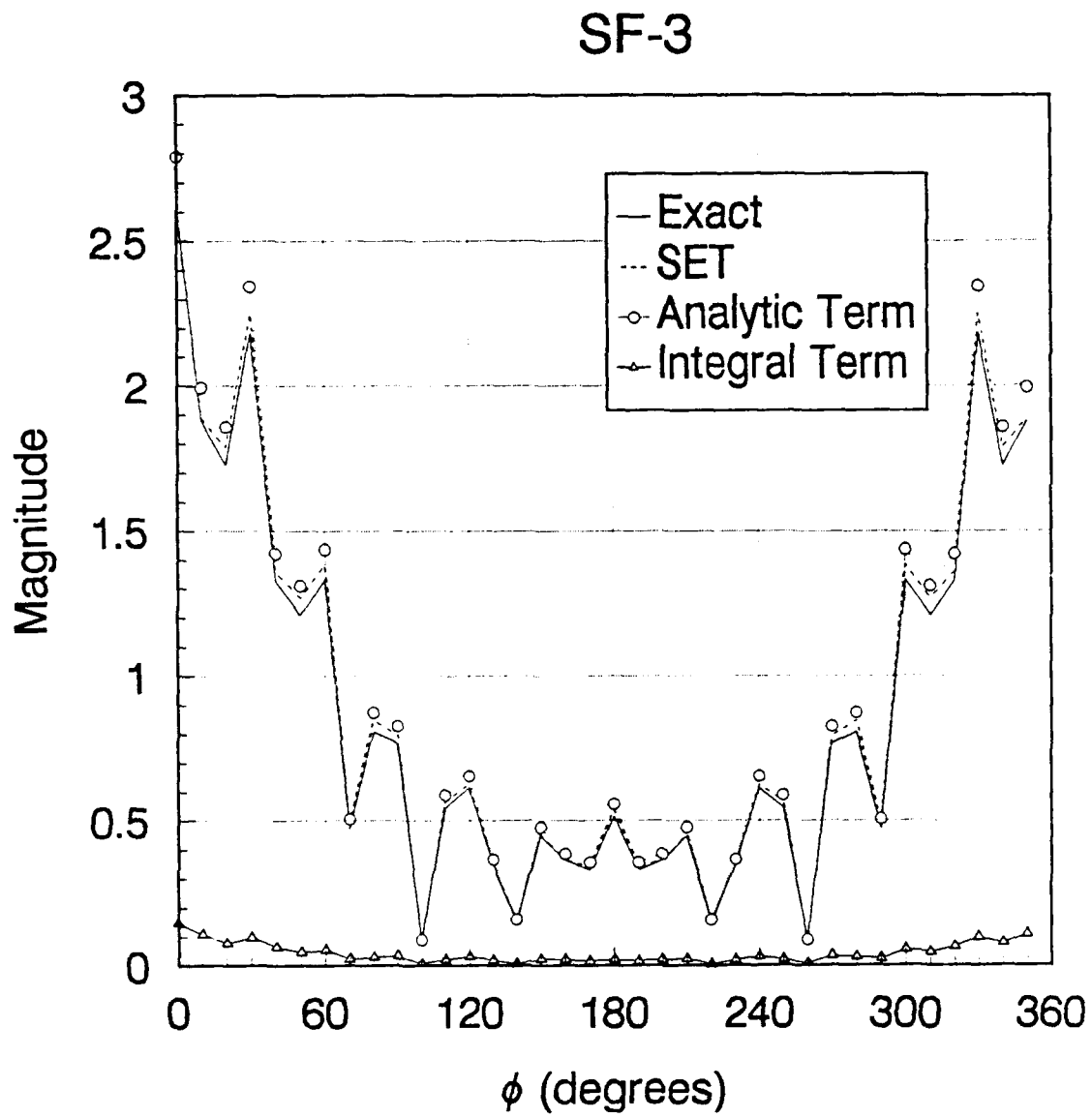


Figure 18. Near-Field for Circular Cylinder, Scattered Field Integration

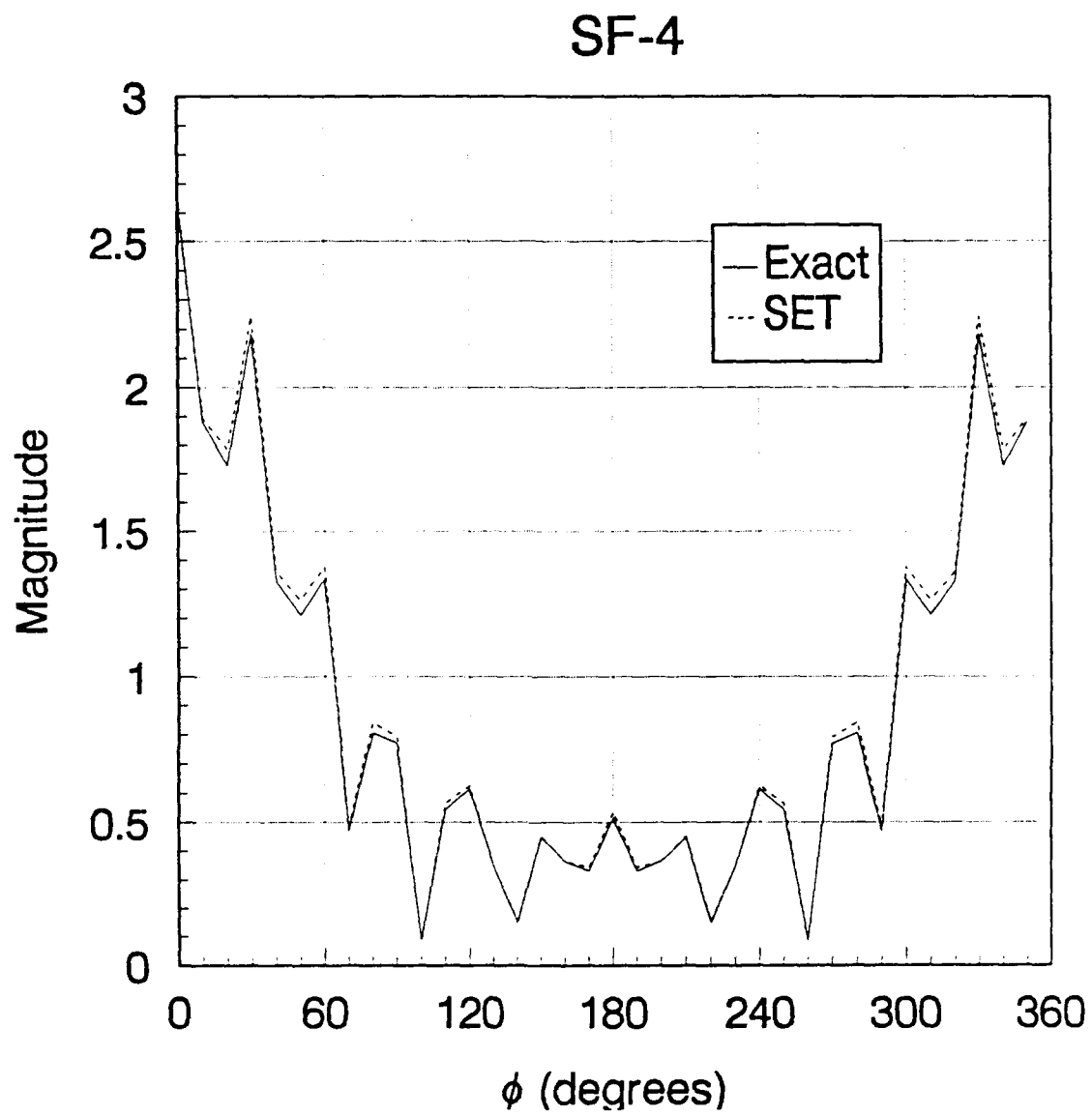


Figure 19. Near-Field for Circular Cylinder, Scattered Field Integration

SF-5

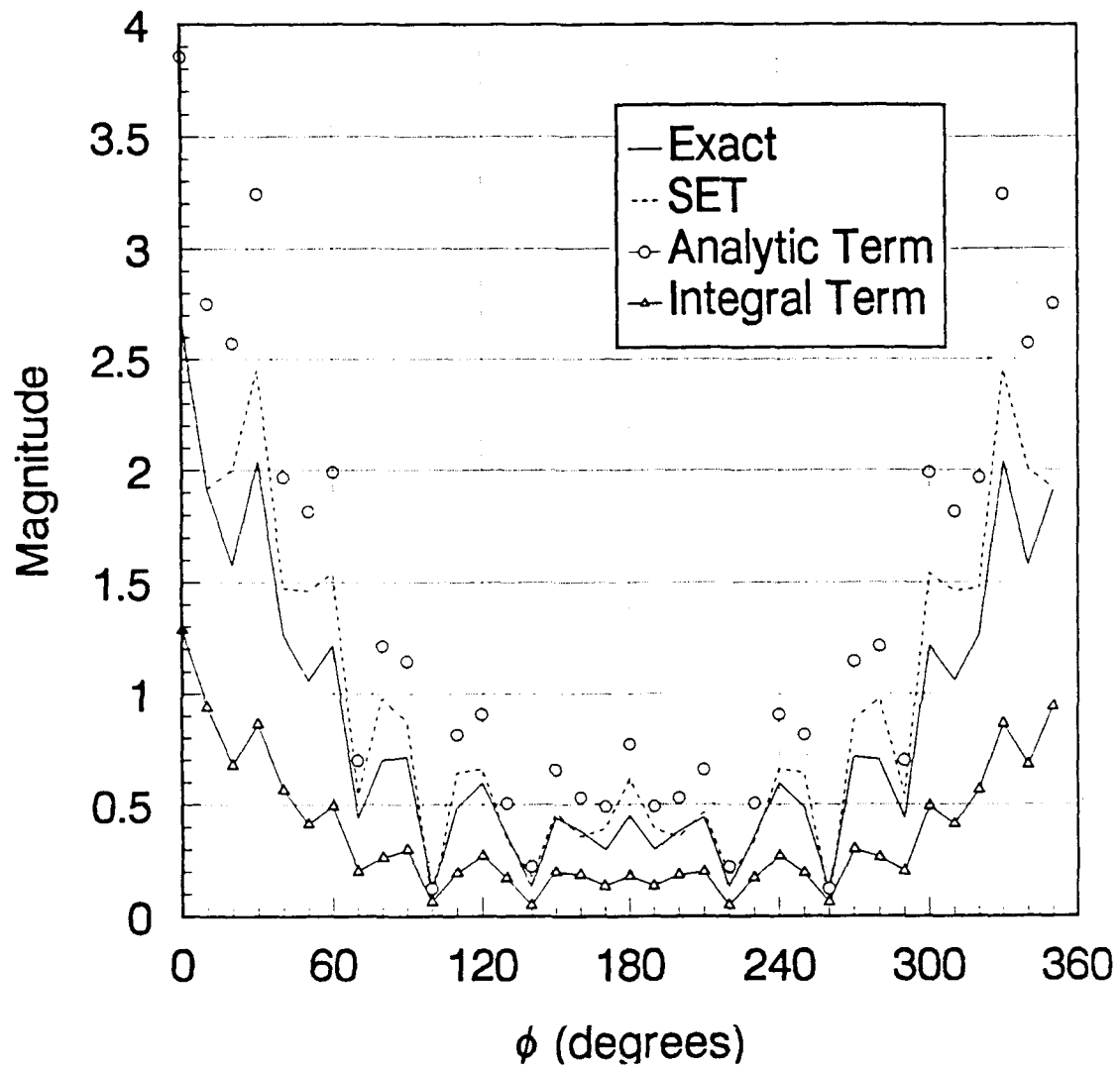


Figure 20. Near-Field for Circular Cylinder, Scattered Field Integration

SF-6

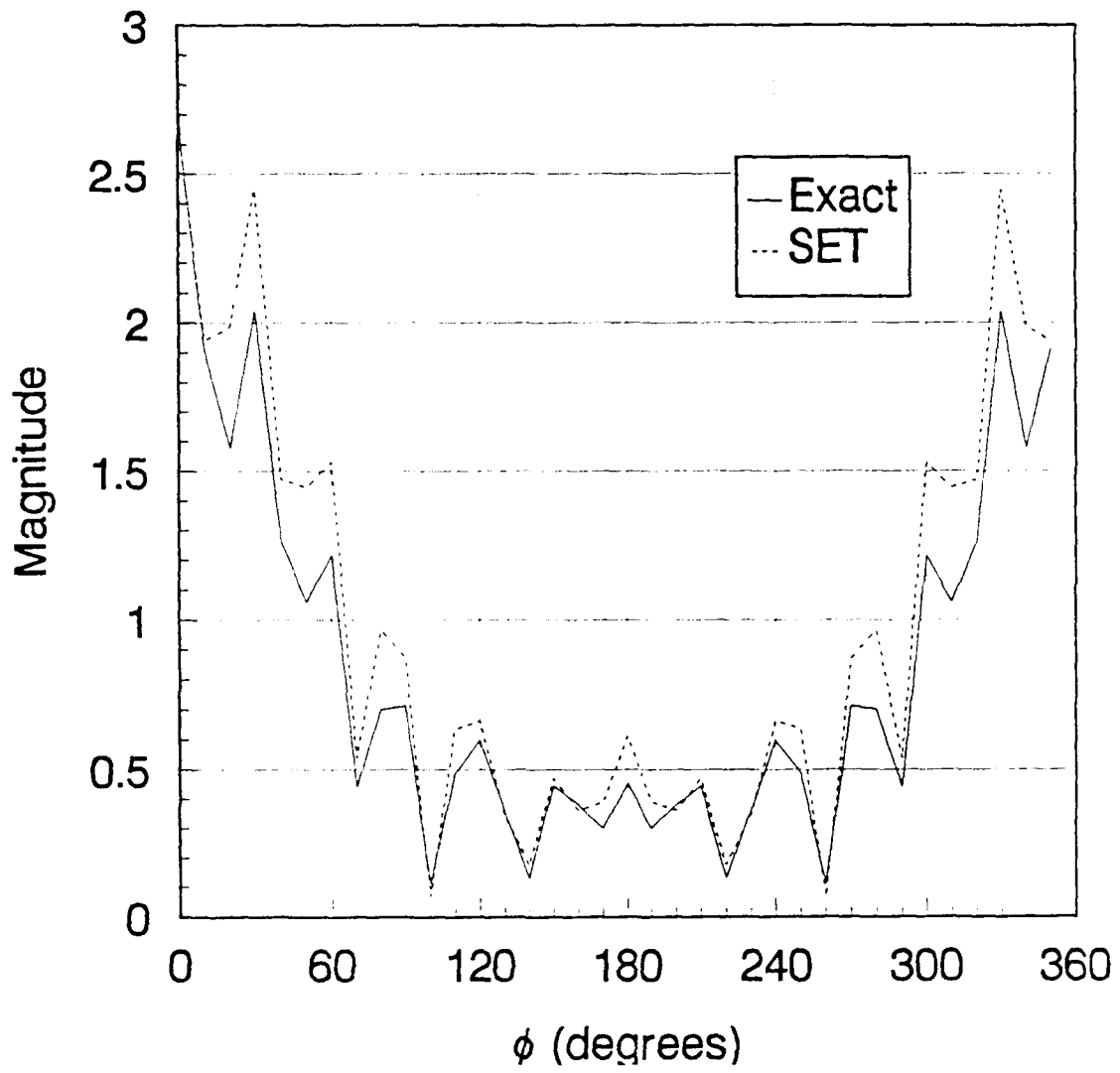


Figure 21. Near-Field for Circular Cylinder, Scattered Field Integration

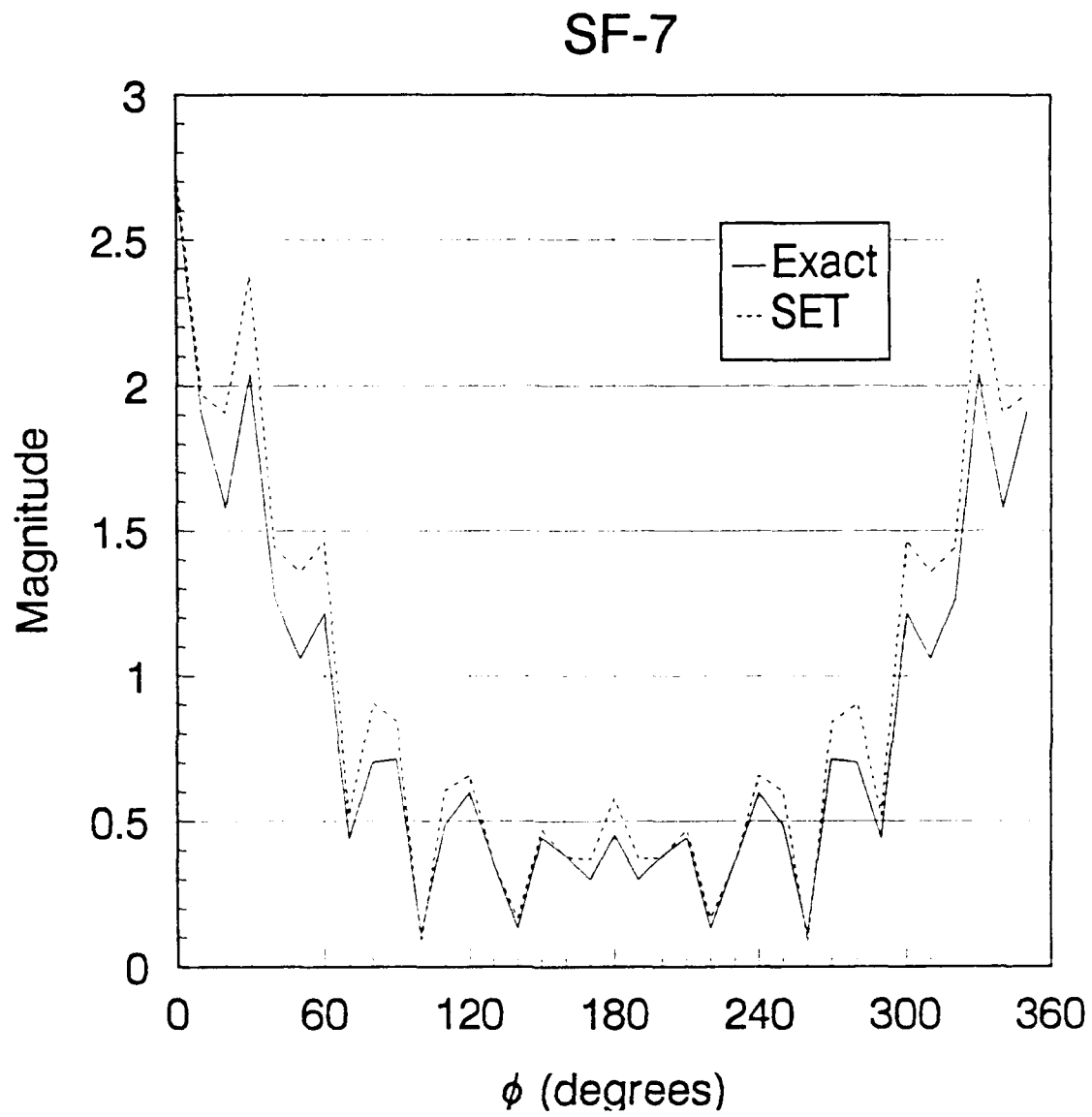


Figure 22. Near-Field for Circular Cylinder, Scattered Field Integration

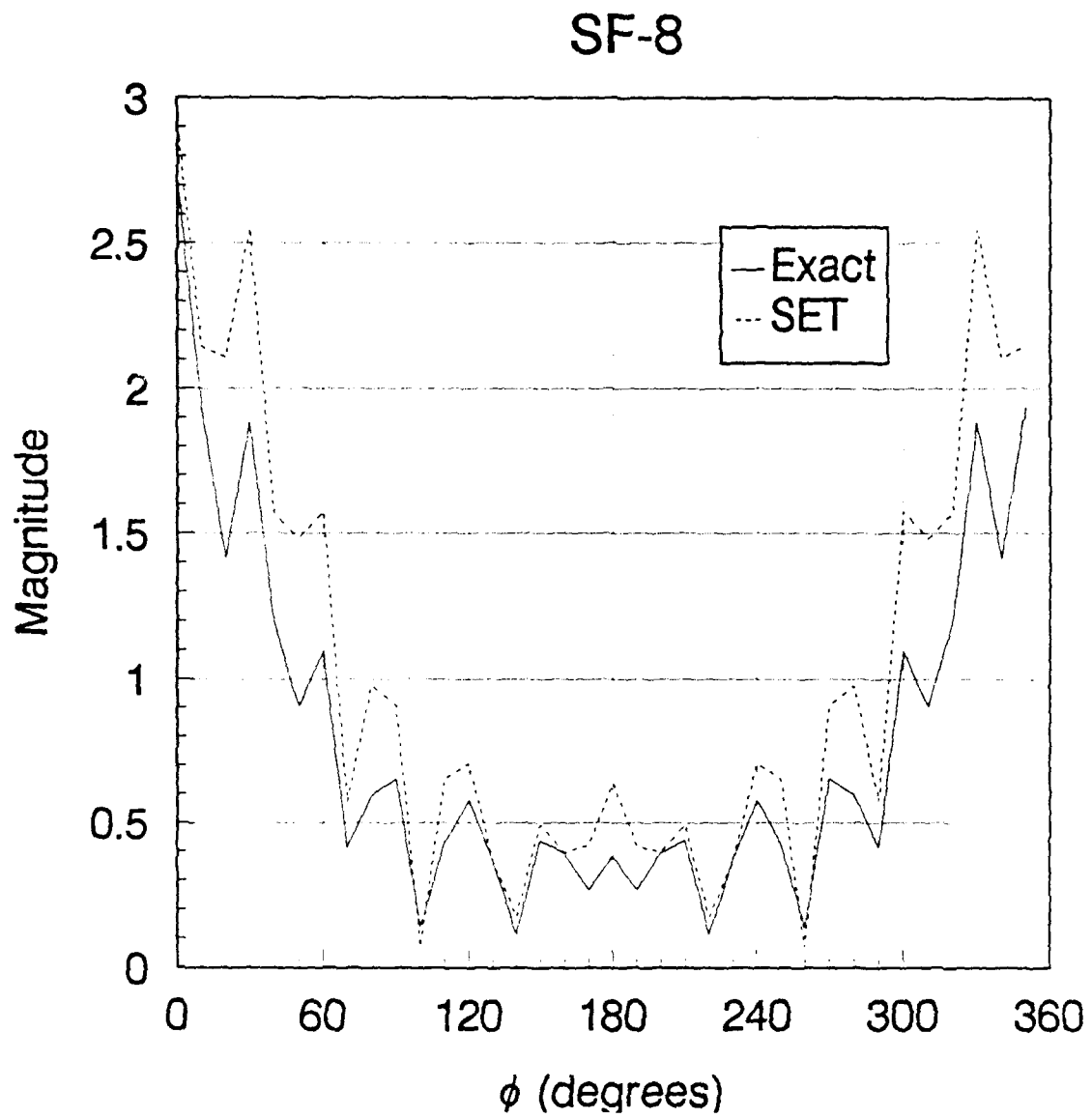


Figure 23. Near-Field for Circular Cylinder, Scattered Field Integration

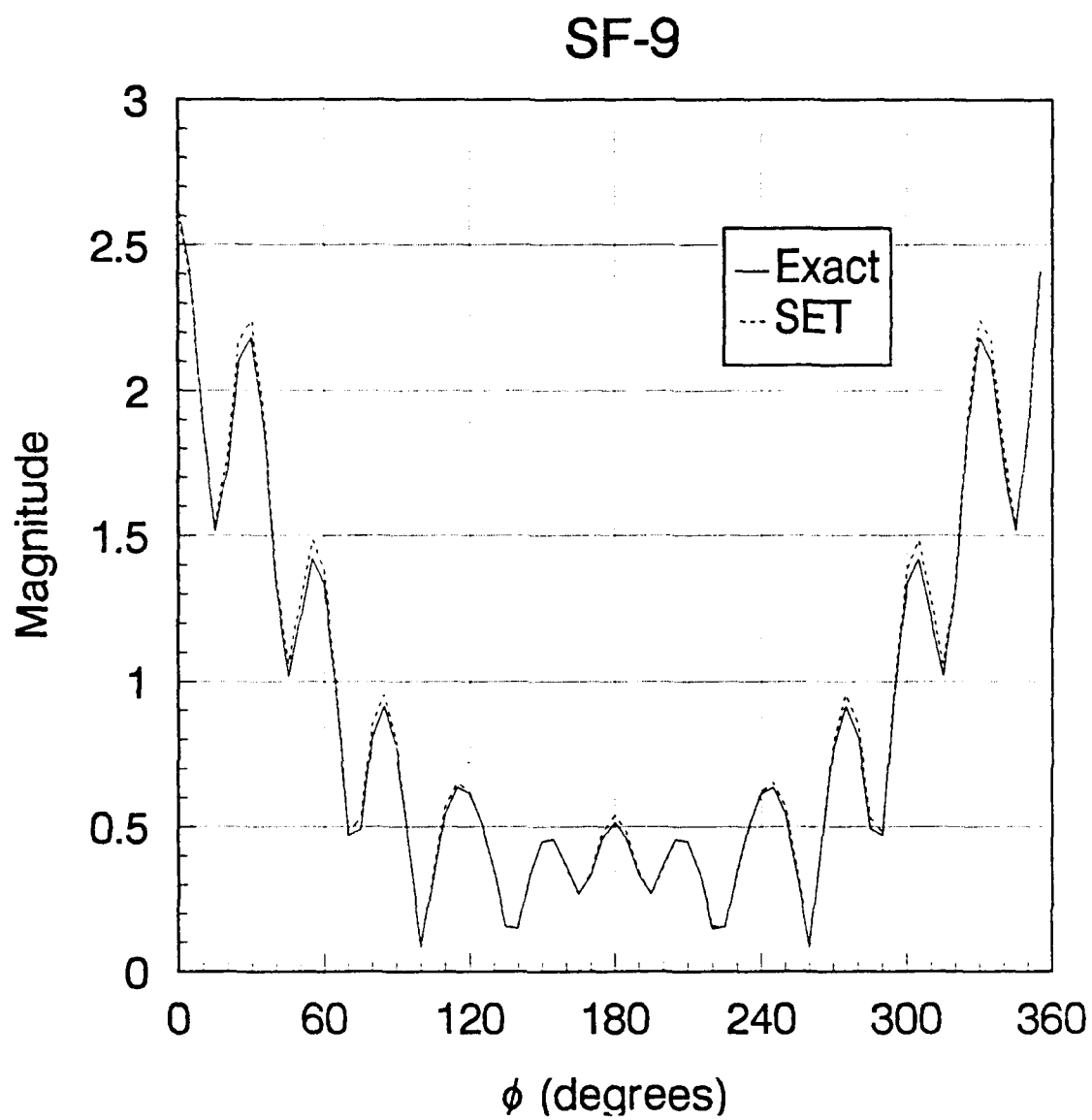


Figure 24. Near-Field for Circular Cylinder, Scattered Field Integration

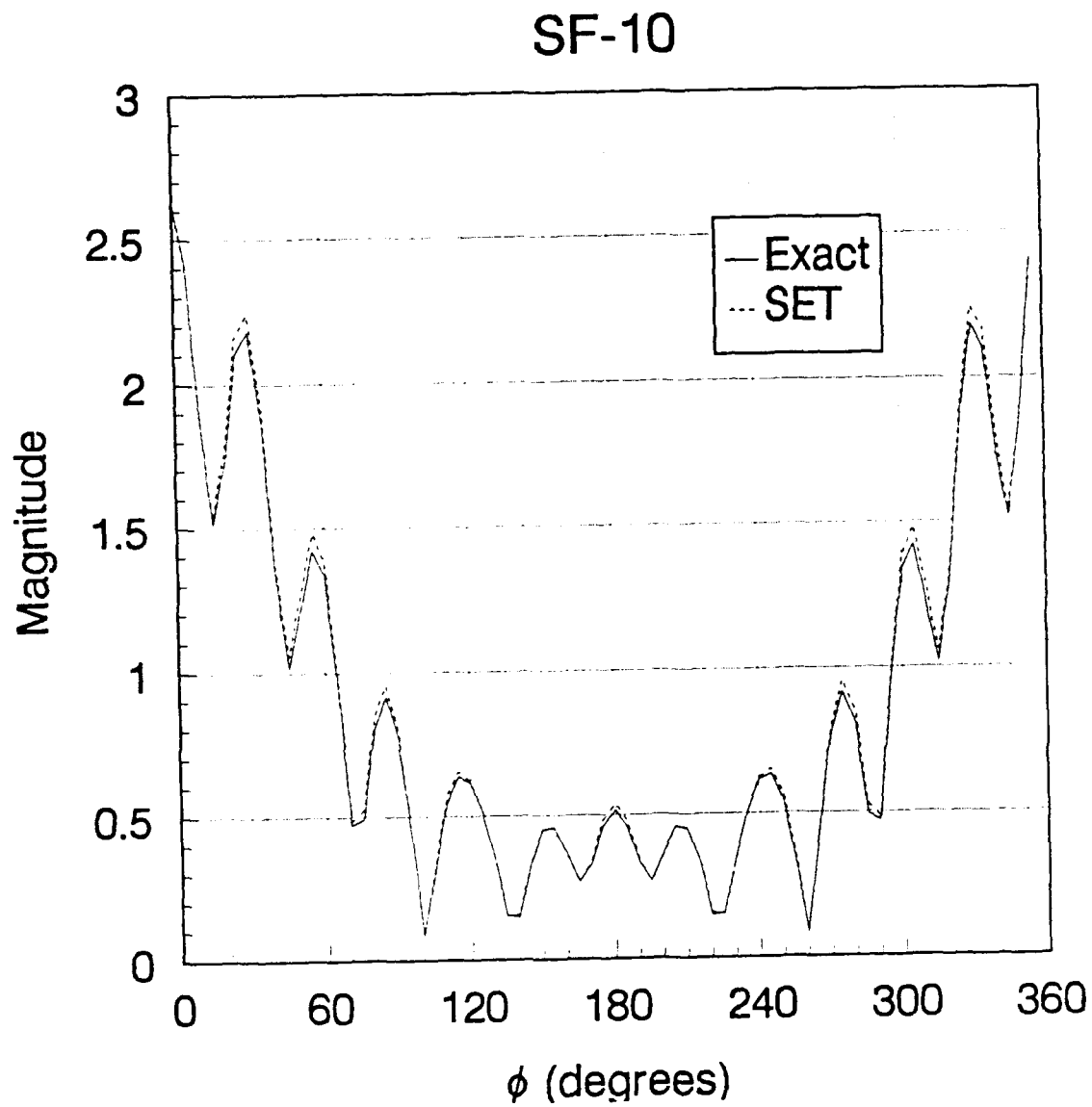


Figure 25. Near-Field for Circular Cylinder, Scattered Field Integration

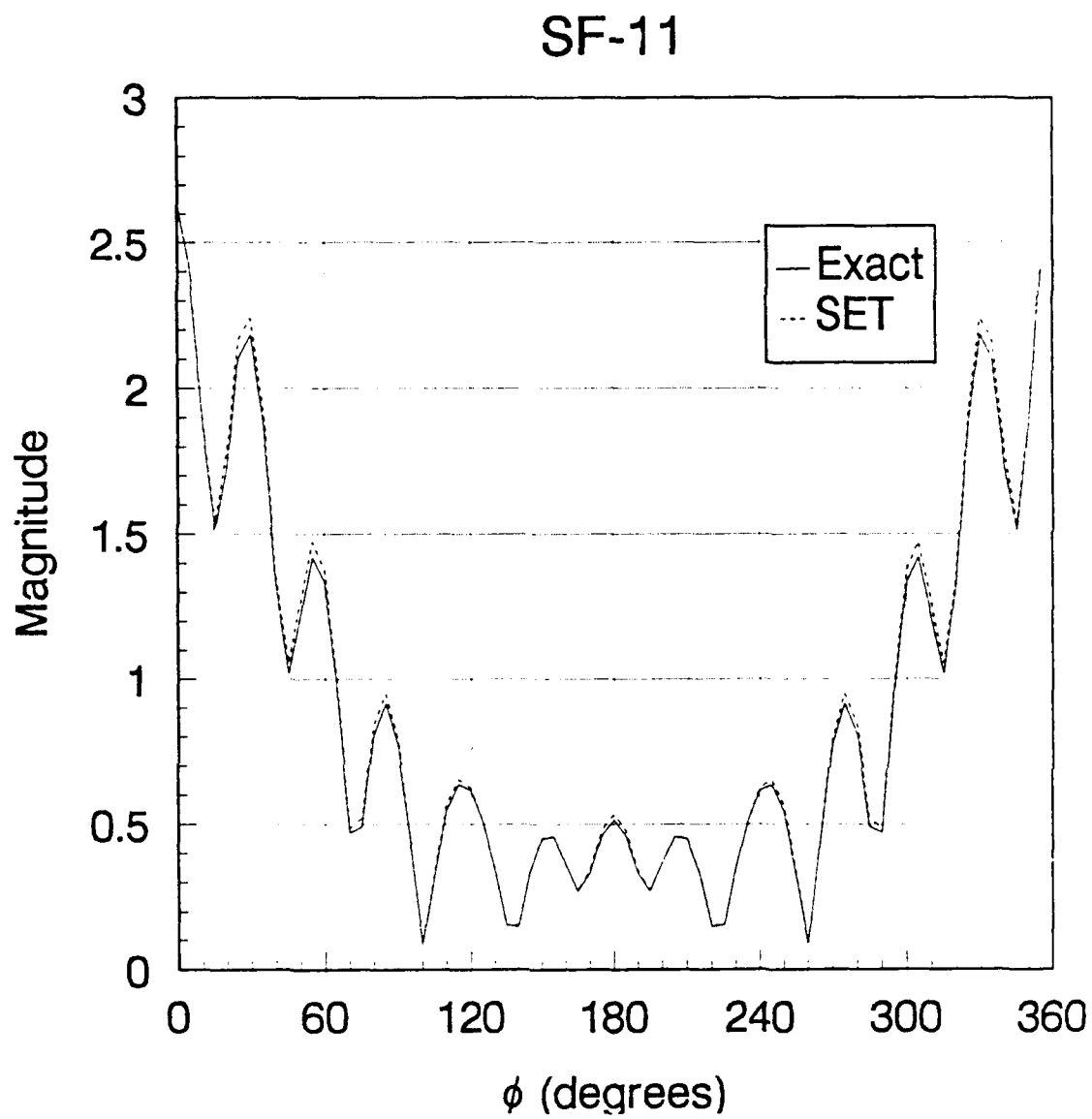


Figure 26. Near-Field for Circular Cylinder, Scattered Field Integration

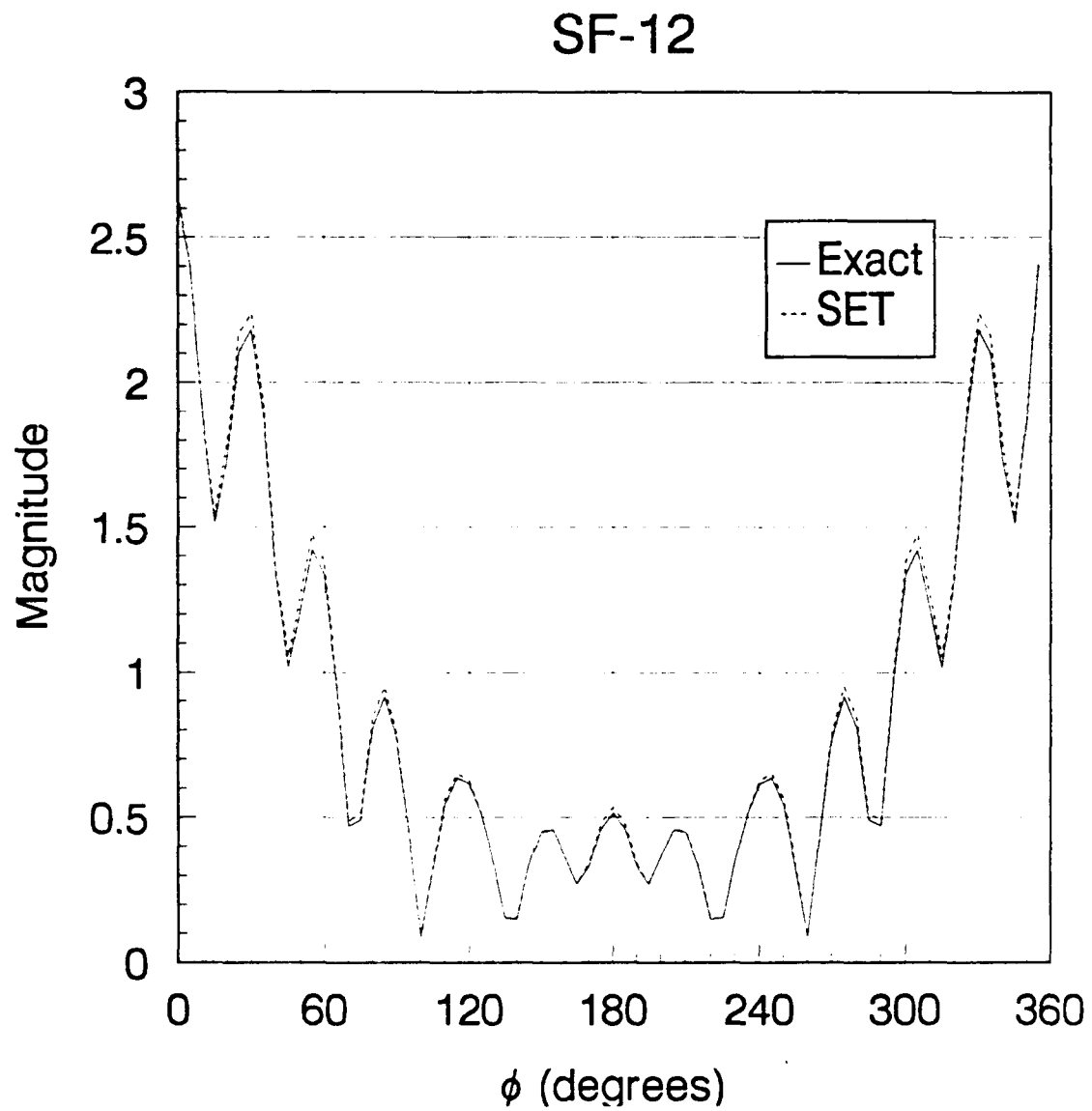


Figure 27. Near-Field for Circular Cylinder, Scattered Field Integration

good results. All four cases, however, exhibit improvement over the corresponding test cases appearing in Figures 16-19 which use fewer node points. This clearly demonstrates the importance of the sampling rate requirement.

The effects of increased frequency are considered next. As mentioned previously, increased frequency has the effect of increasing the electrical length of the object perimeter, thus requiring more sampling nodes. Three different sampling rates were considered in these tests. First, an undersampled case was examined with a sampling rate of less than 1.5 samples per cycle which produced extremely inaccurate results as illustrated in Figure 28. Increasing the sampling rate has a beneficial effect on the solution as seen in Figure 29, but the desired accuracy is still lacking. A sufficient number of samples (approximately 6 per cycle) were taken for the case depicted in Figure 30 producing an extremely accurate near-field solution for the high frequency case.

Changing the relative permittivity or permeability should have an effect on the near field similar to that of frequency. Increased ϵ_r or μ_r should require more nodes, or a higher sampling rate to accurately represent the near-field. Four test cases were considered with $\epsilon_r = \mu_r = 5$. Figure 31 represents the case with the fewest nodes. The sampling rate was increased in Figures 32-34. Initially, it appears that the low sampling rate produced the more accurate near-field. However, comparisons at specific points on the boundary contour indicate that a higher sampling rate yields the more accurate results.

SF-13

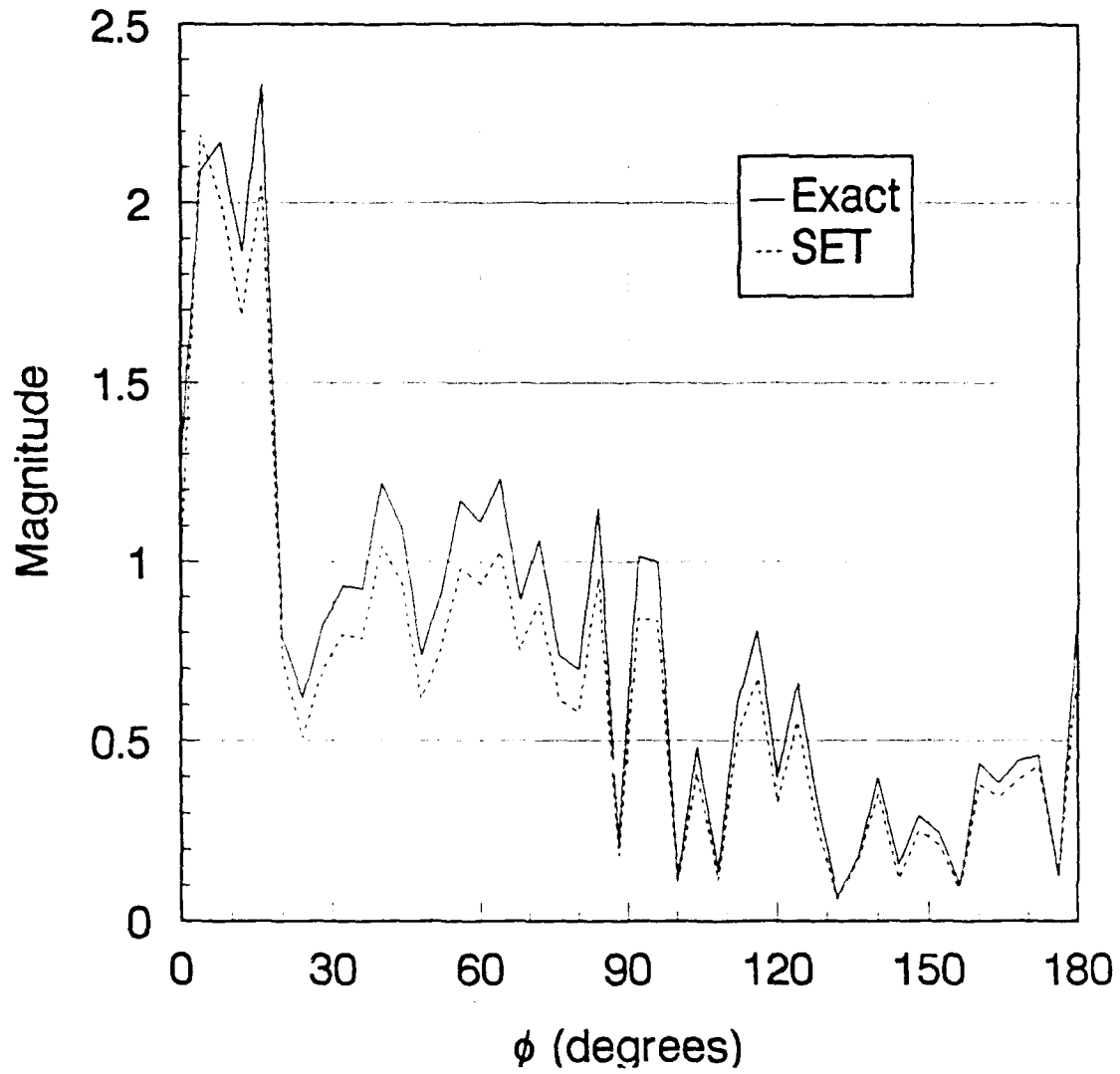


Figure 28. Near-Field for Circular Cylinder, Scattered Field Integration

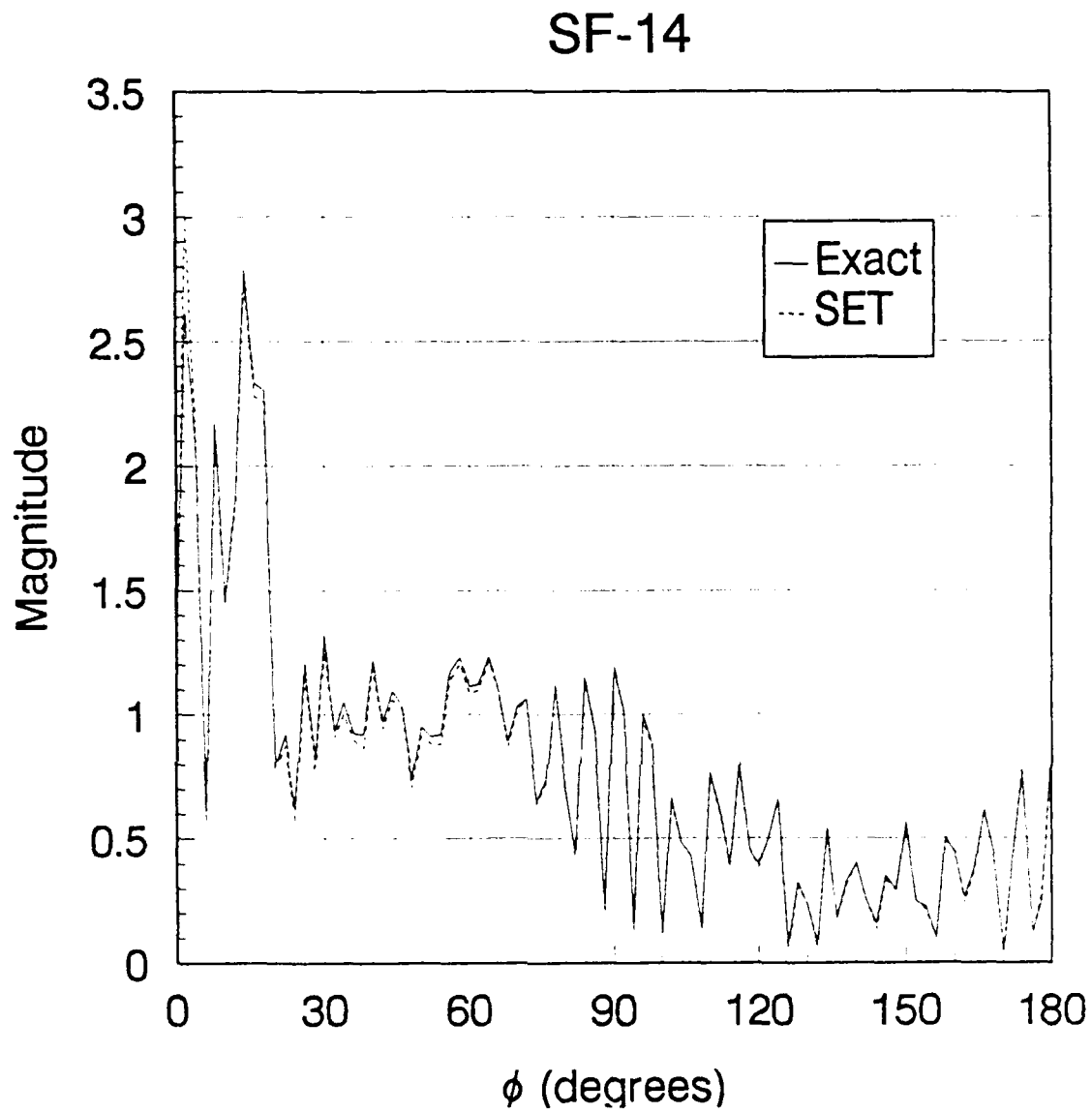


Figure 29. Near-Field for Circular Cylinder, Scattered Field Integration

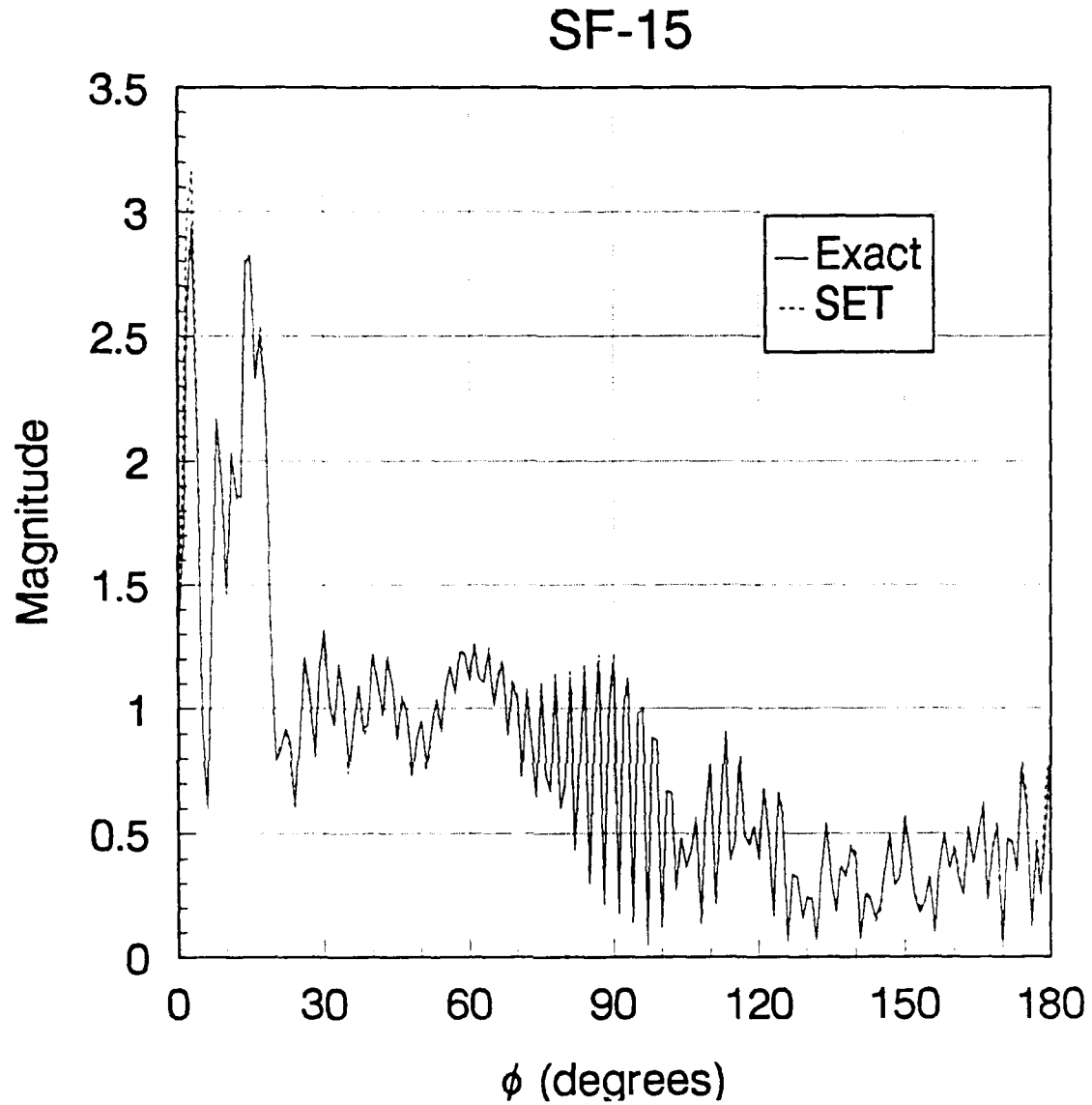


Figure 30. Near-Field for Circular Cylinder, Scattered Field Integration

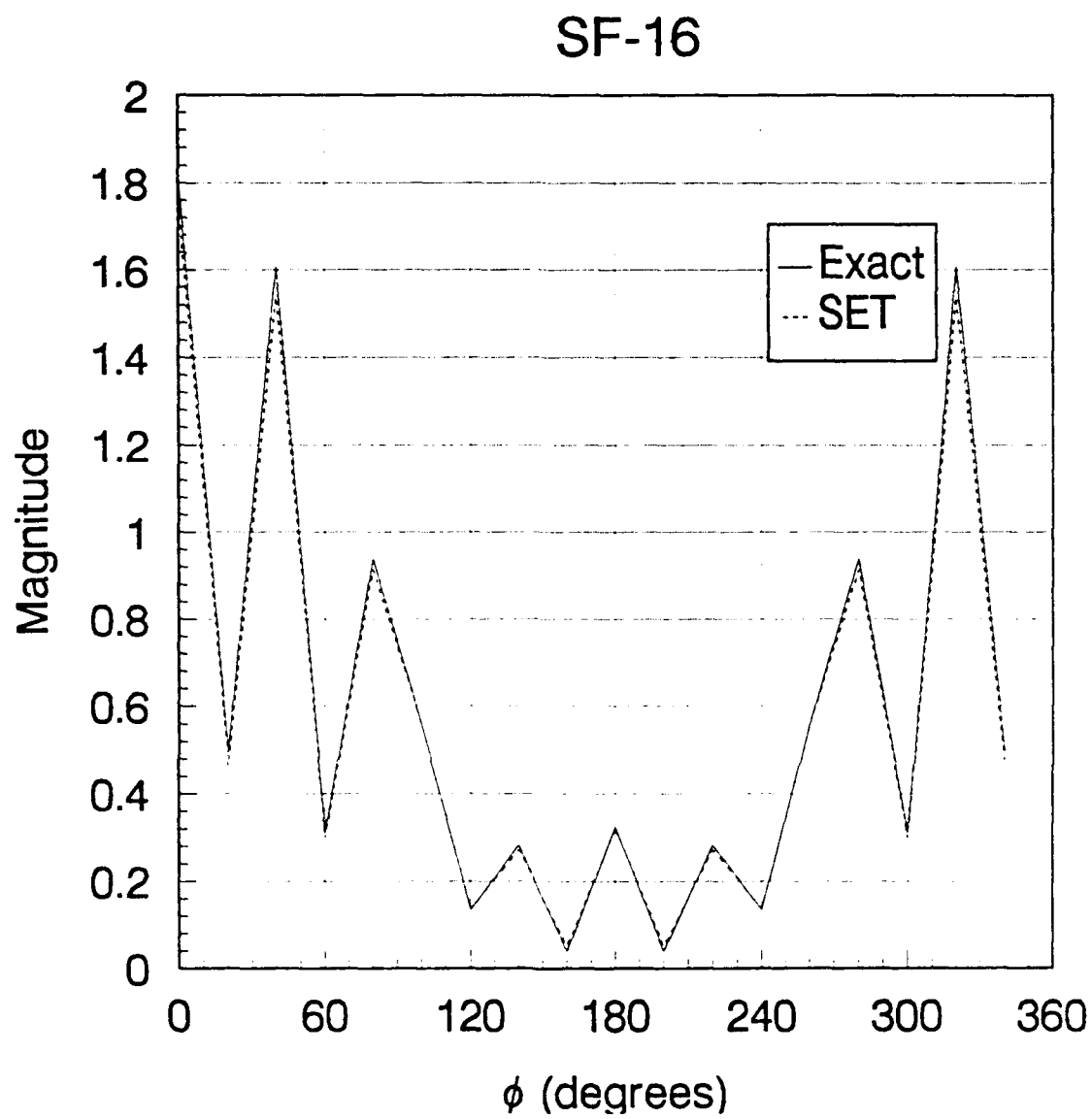


Figure 31. Near-Field for Circular Cylinder, Scattered Field Integration

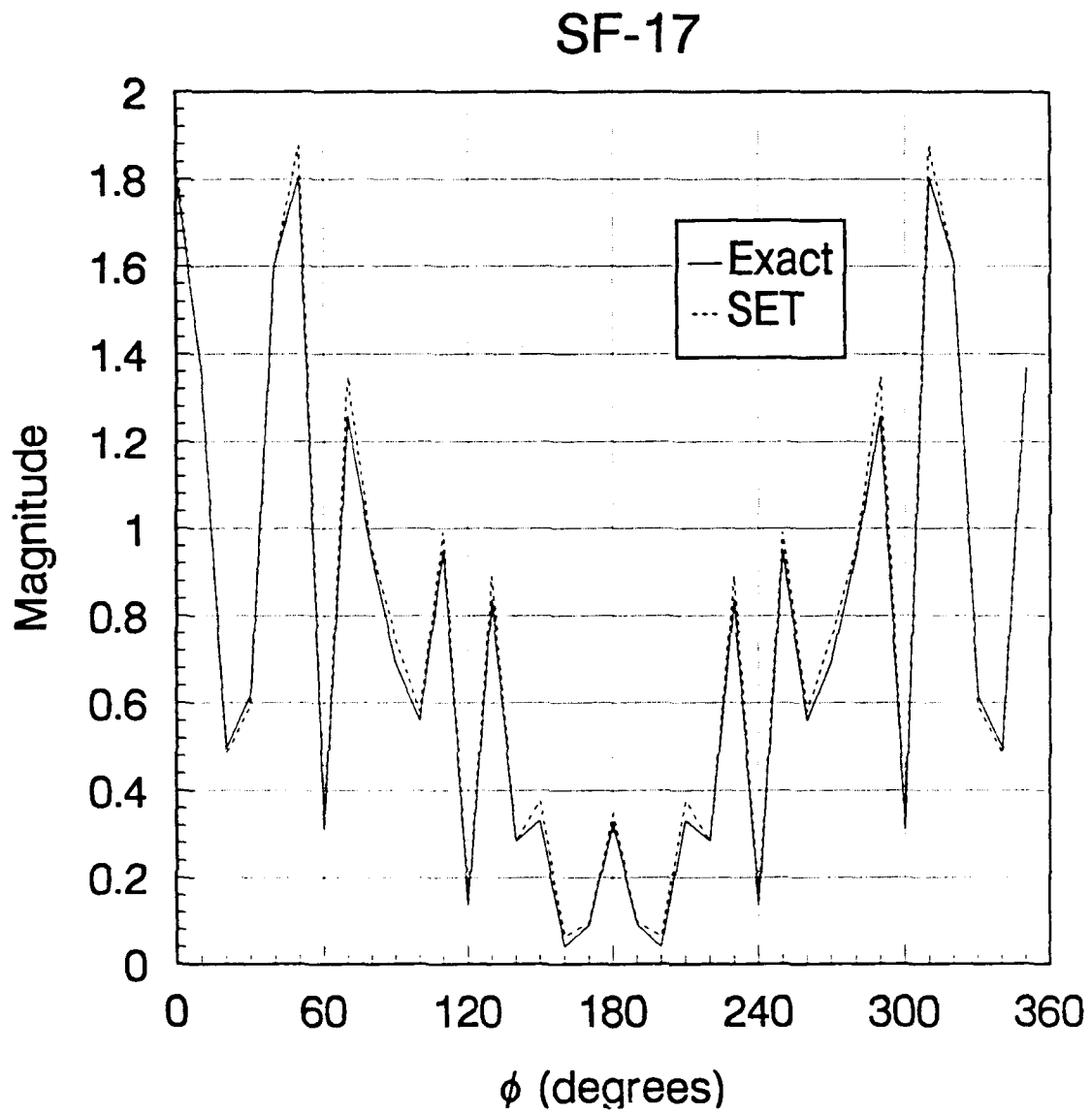


Figure 32. Near-Field for Circular Cylinder, Scattered Field Integration

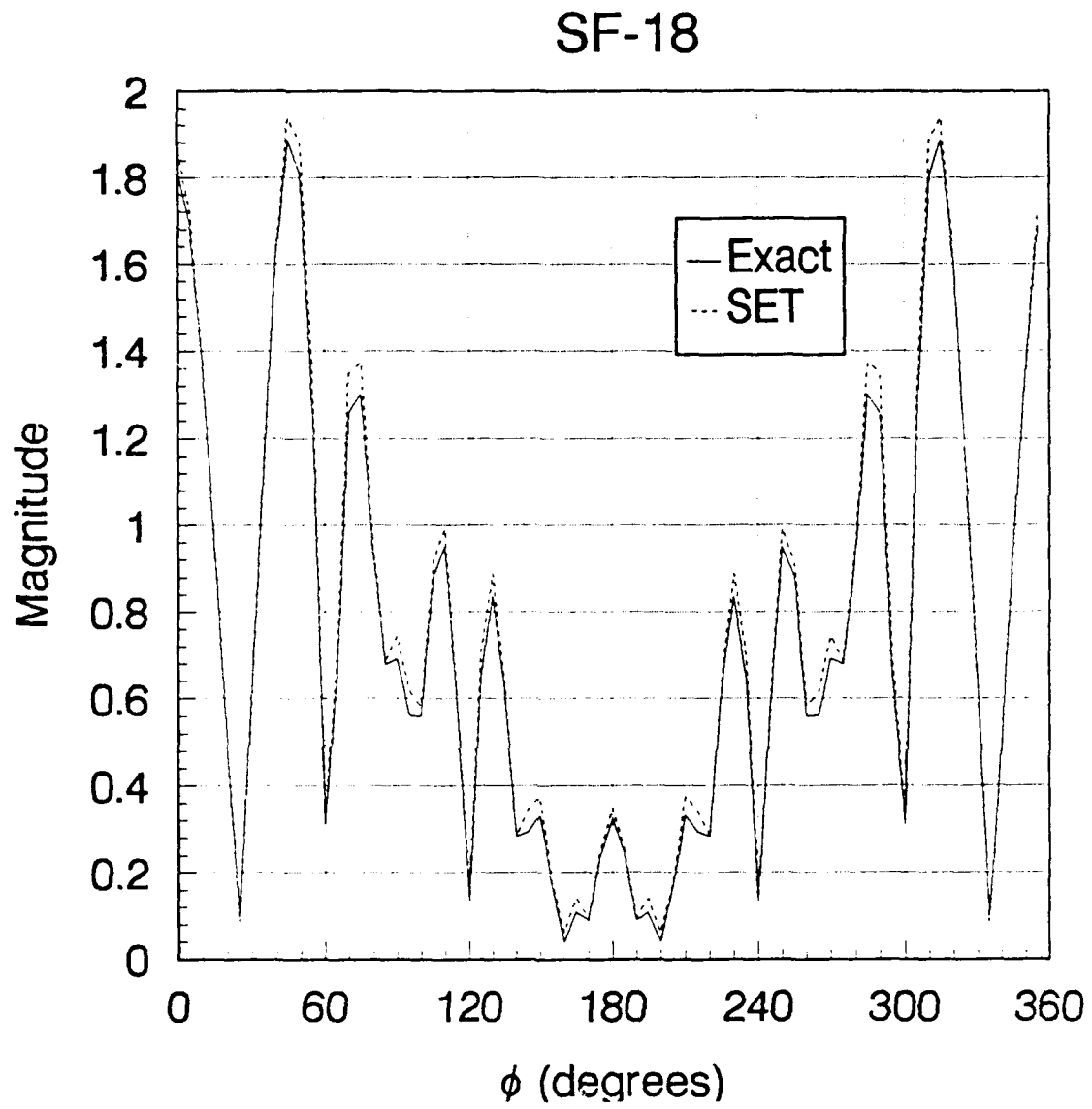


Figure 33. Near-Field for Circular Cylinder, Scattered Field Integration

SF-19

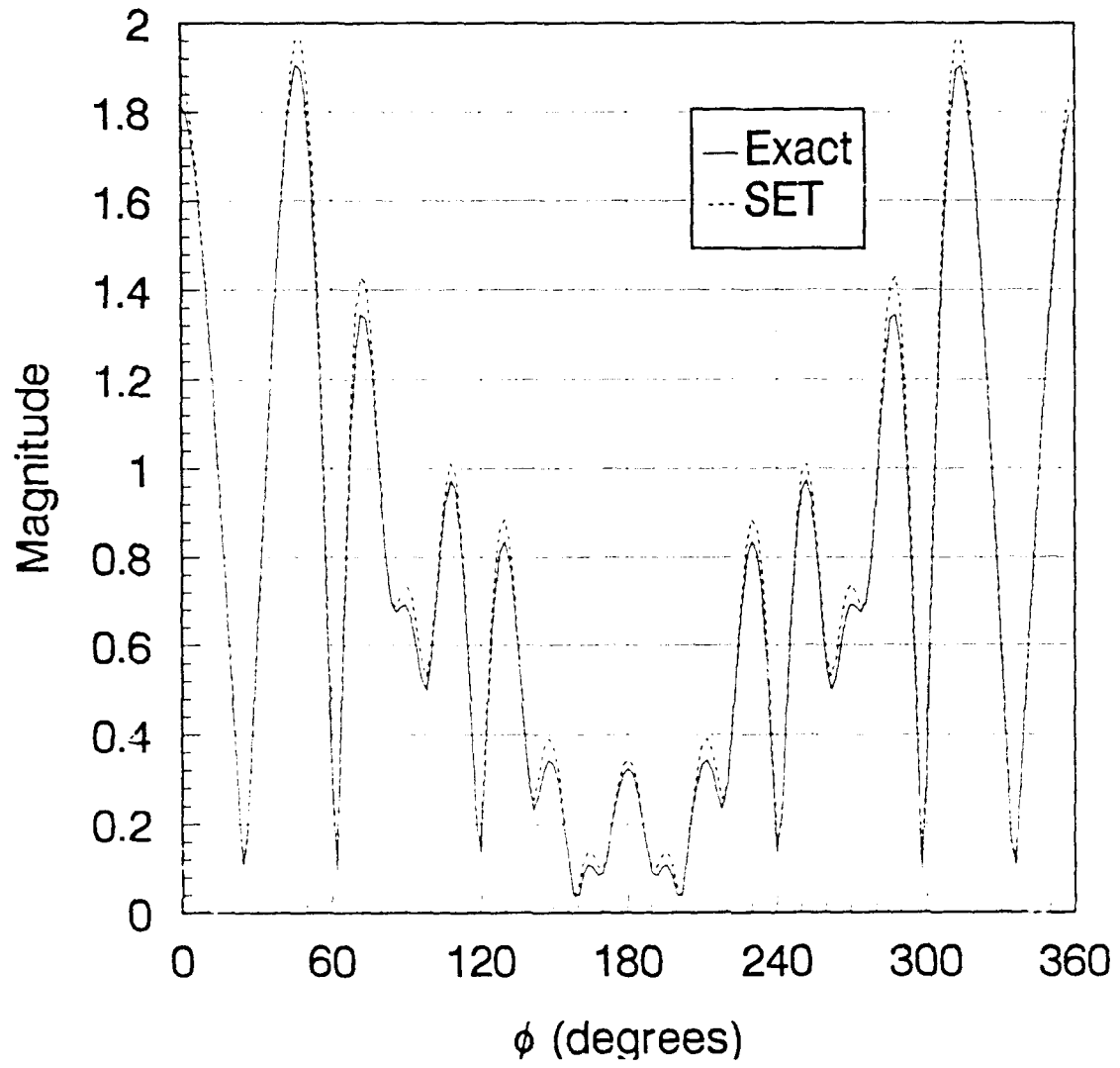


Figure 34. Near-Field for Circular Cylinder, Scattered Field Integration

Next, a comparison of Equations (20) and (21) was made. The near-field calculations should be identical for both forms of the SET. Recall that Equation (20) uses scattered field inside the integral, whereas Equation (21) uses the total field. Figures 35-38 depict the results for four cases, each calculated using both equations. As seen in the figures, the results from both equations are almost identical for each case considered.

In order to make relative comparisons of the test cases above, a quantitative description of the accuracy was required. The relative error function,

$$\gamma = \frac{\sum_1^N |\psi_{Exact}^{(s)} - \psi_{SET}^{(s)}|}{\sum_1^N |\psi_{Exact}^{(s)}|}, \quad (32)$$

was used to establish a representative quantity to be used in comparisons of characteristic cases. The relative errors for several cases considered above were calculated for comparison. Table 5 lists the relative error calculated for the cases depicted in Figures 16, 19, 24, 25, and 30. The relative error is very small in all cases indicating good agreement of the exact and SET solutions.

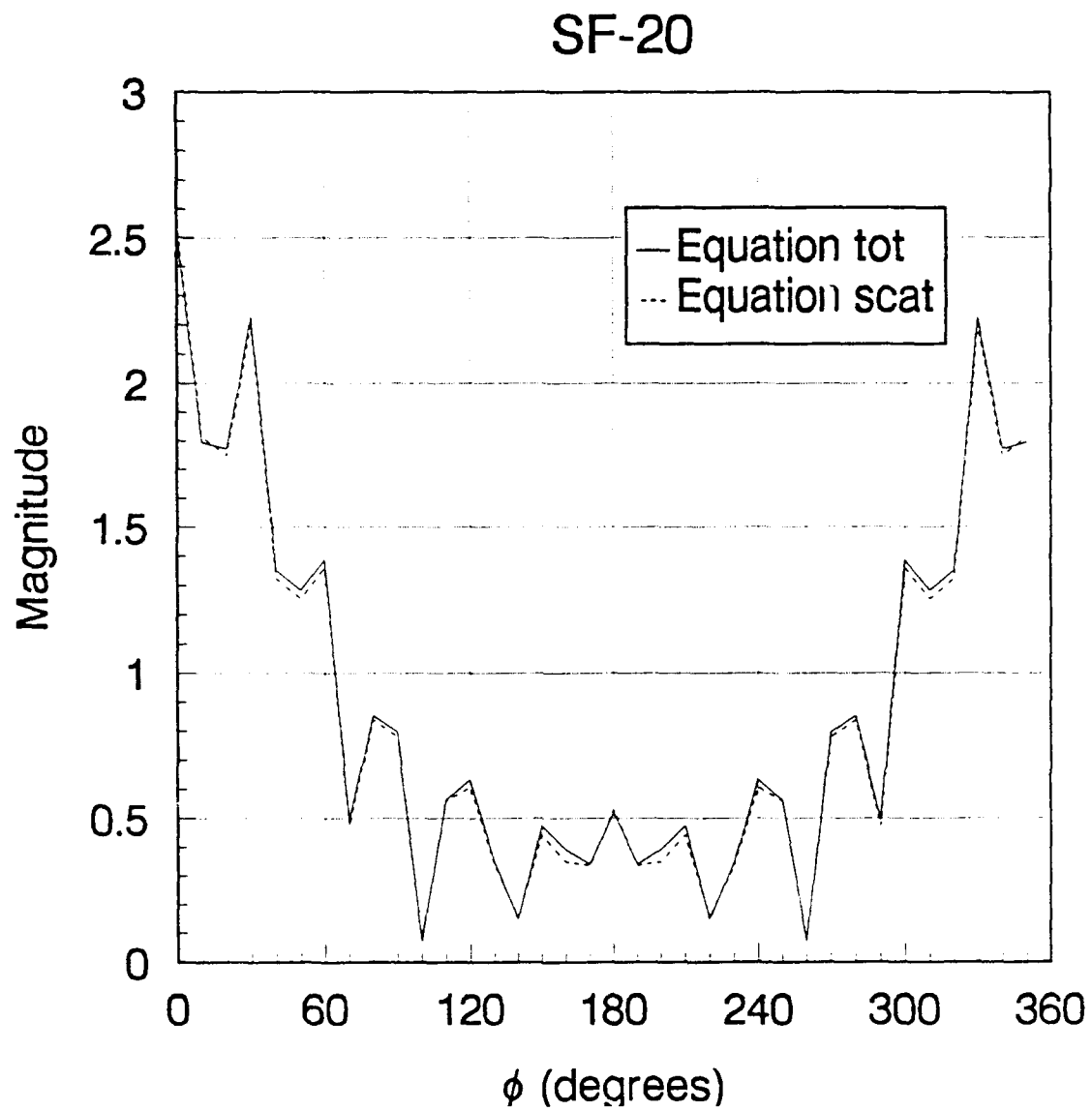


Figure 35. Near-Field for Circular Cylinder, Total Field Integration

SF-21

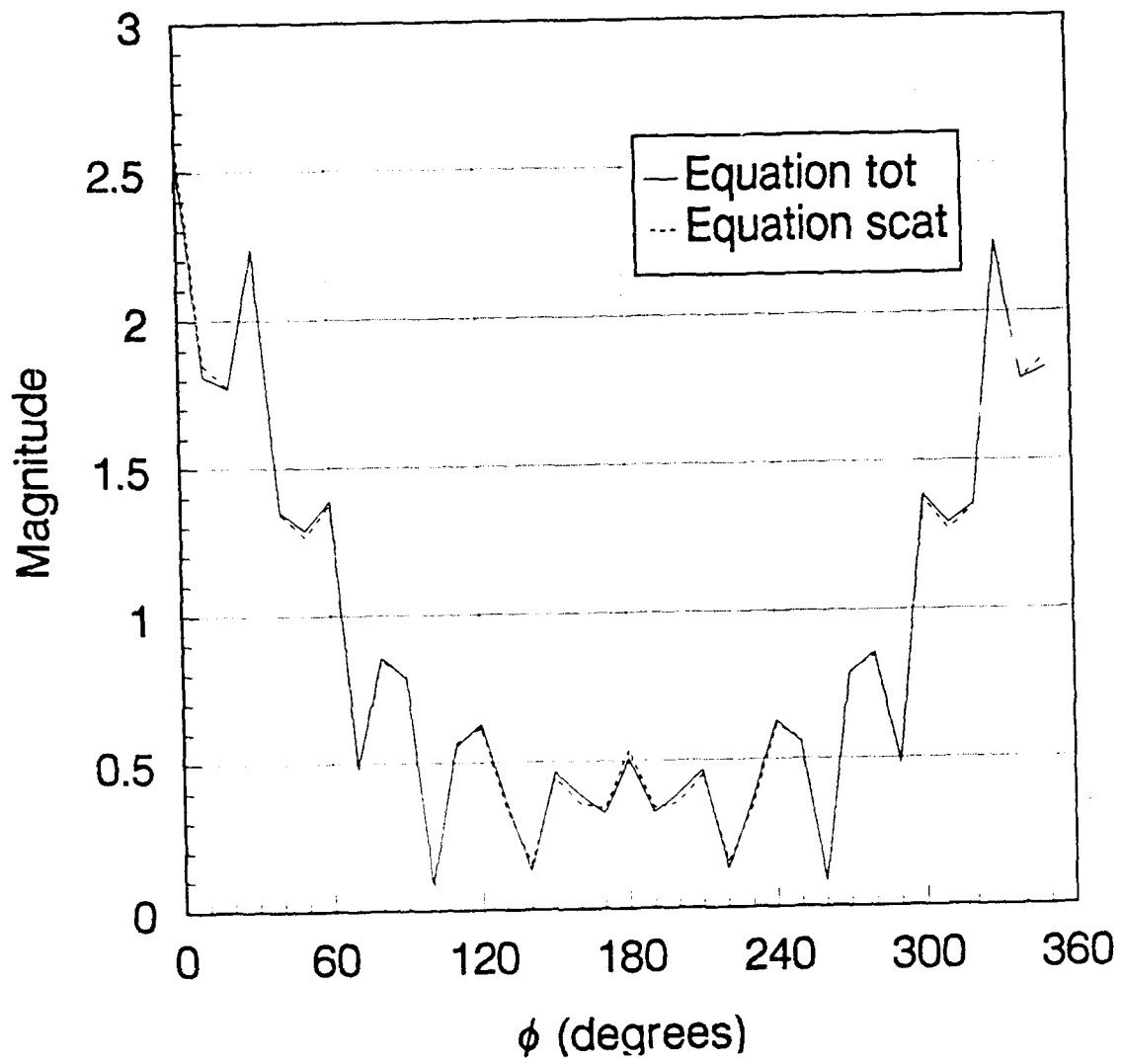


Figure 36. Near-Field for Circular Cylinder, Total Field Integration

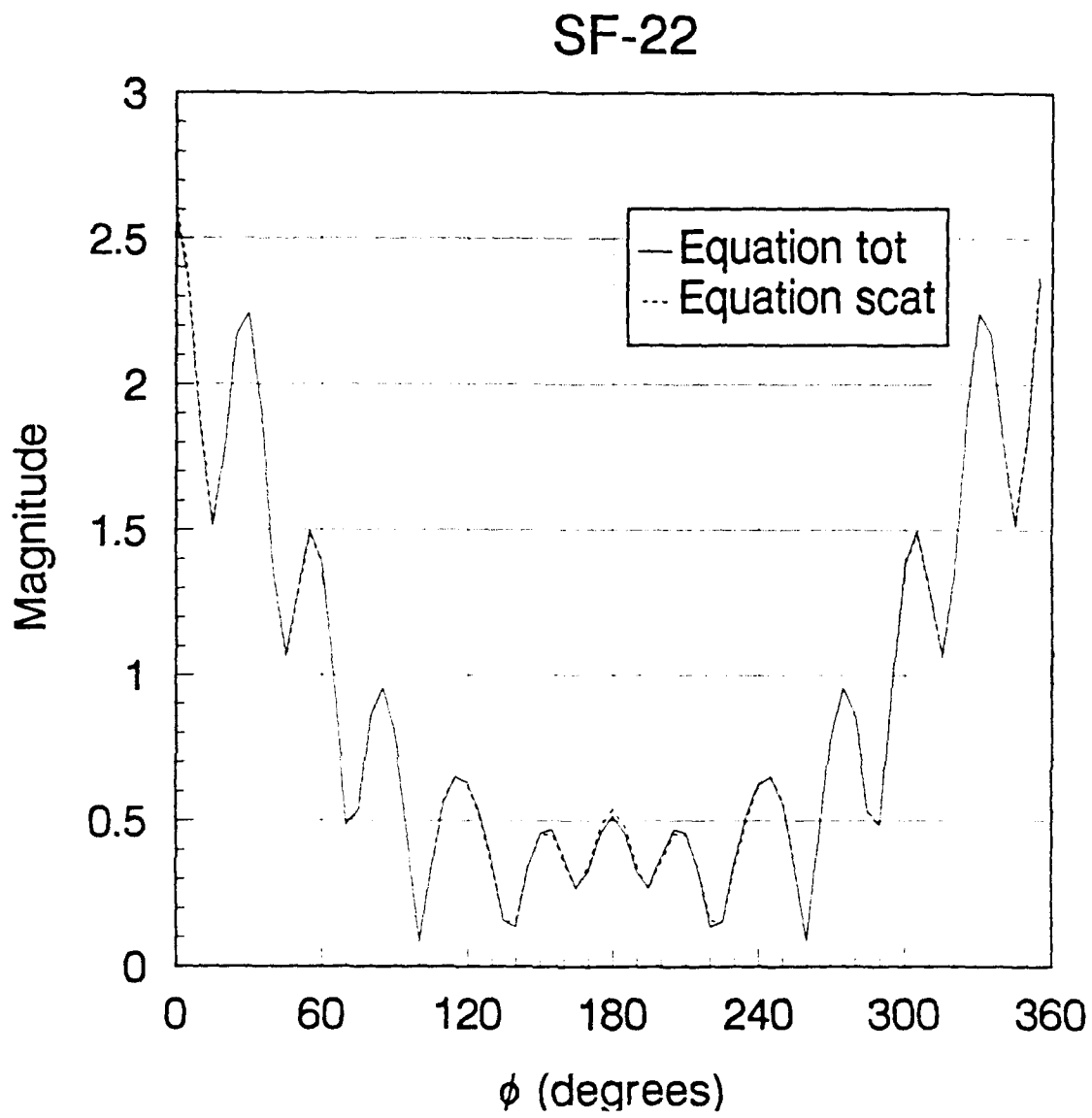


Figure 37. Near-Field for Circular Cylinder, Total Field Integration

SF-23

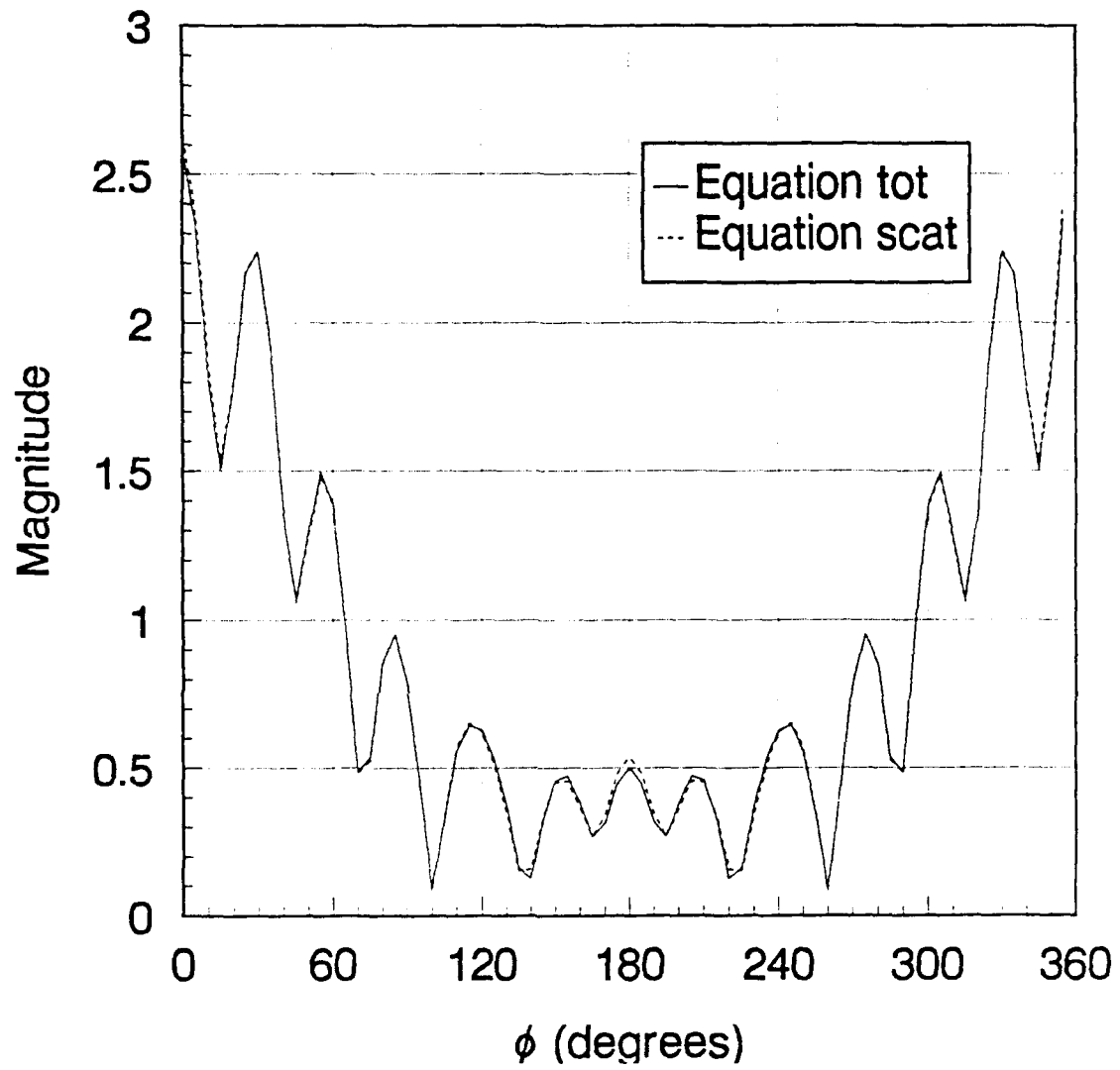


Figure 38. Near-Field for Circular Cylinder, Total Field Integration

TABLE 5. ACCURACY OF SET

CASE	γ
SF-1	0.064
SF-2	0.057
SF-9	0.065
SF-10	0.058
SF-15	0.058

E. TIMED EVALUATIONS

The last phase of testing and validation were time test. Benchmark elapsed times were established for representative cases. Elapsed time, as well as accuracy were also observed for situations in which integration in the asymptotic region of the contour is bypassed. The integral in Equation (20) is bypassed for source points greater than EPS1 away from the field point (i.e., $k_0 R > \text{EPS1}$). Two typical cases were evaluated for various EPS1.

Table 6 illustrates the sharp decrease in elapsed run time when the integration routine is bypassed in the asymptotic region. However, the accuracy of the near-field calculation is extremely degraded as depicted in Figures 39 and 40.

TABLE 6. SET ELAPSED TIME

CASE	EPSI	TIME (h:m:s)
SF-24	SET	42:05
	0.4π	2:35
	0.6π	3:47
	π	6:12
SF-25	SET	3:06:10
	4π	11:28
	6π	17:46
	π	29:06

* SET - EPSI bypass not invoked

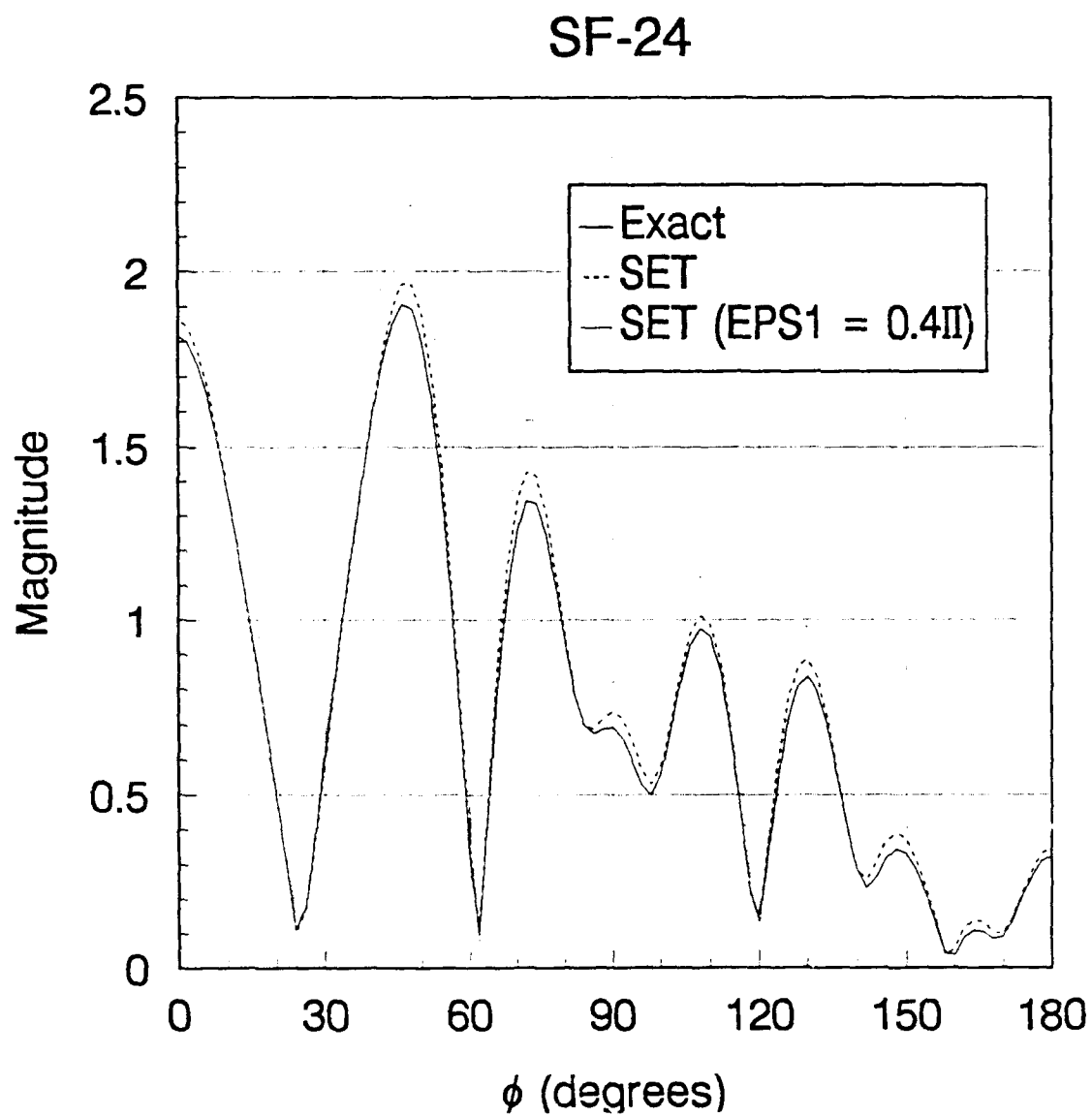


Figure 39. Near-Field for Circular Cylinder, Asymptotic Contribution Neglected

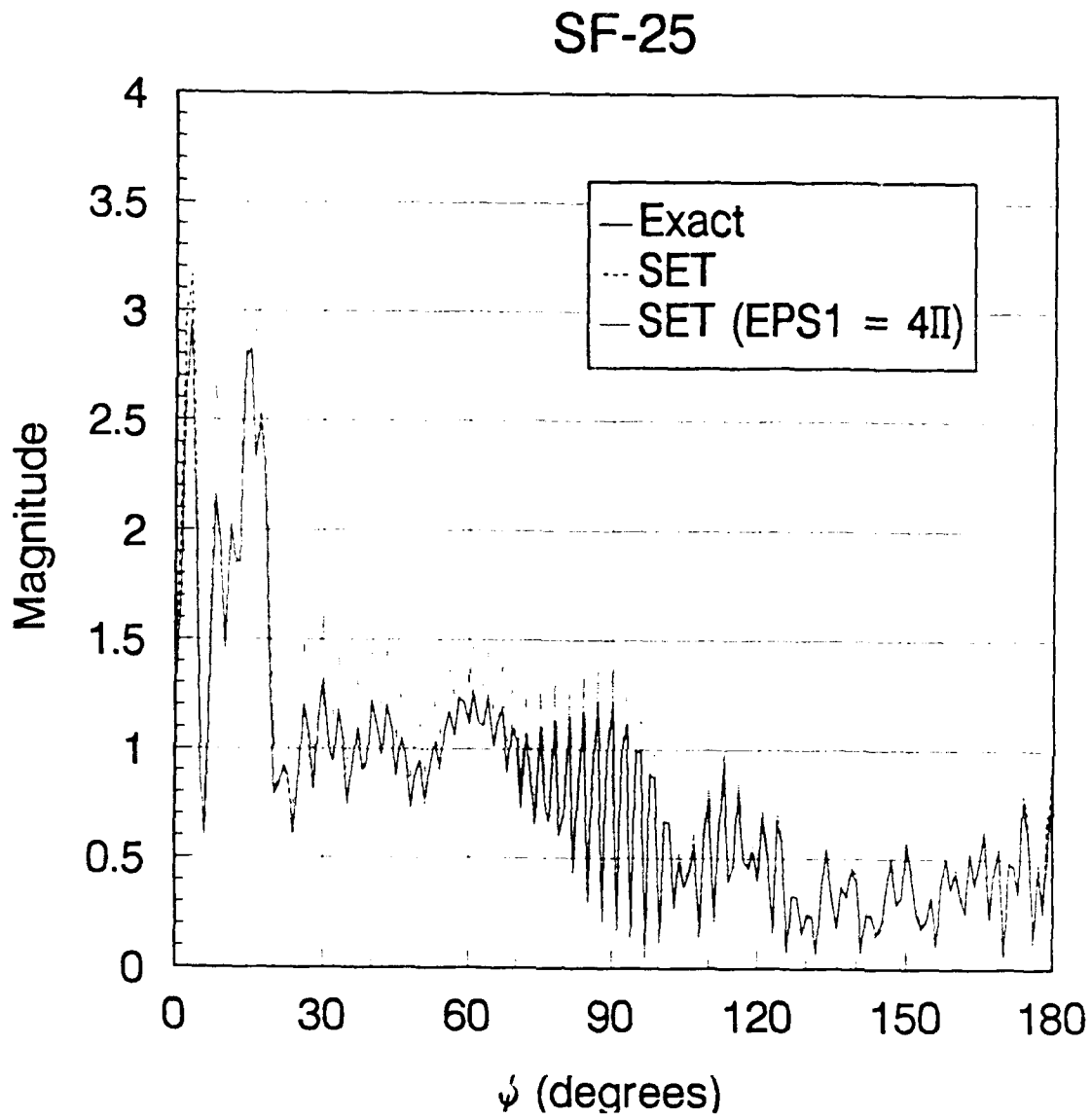


Figure 40. Near-Field for Circular Cylinder, Asymptotic Contribution Neglected

VI. CONCLUSIONS

A. RESULTS

The Singularity Extraction Technique proved to be a useful method of evaluating near-fields for specific cases only. The technique did not consistently provide accurate results for all test cases, it however worked quite well under certain conditions.

Results obtained in the case of integration of the incident field on the object surface were acceptable in the medium frequency range ($f = 300$ MHz) only. Observations for other frequencies deviated significantly from theoretical results. An increase in the sampling rate did, however, demonstrate the convergence of the SET.

Numerous tests were conducted for the implementation of Equation (20). Some of the key observations are listed below.

1. The SET closely approximated the exact solution in most cases considered as long as the sampling rate was sufficient and the offset distance remained relatively small.
2. Significant contributions from both terms of Equation (20) were present in most cases considered.
3. A sufficient sampling rate (number of nodes) was more critical for accuracy than the differential element of the numerical integration.
4. Equation (21) produced results equivalent to those of Equation (20).

5. Exclusion of contributions due to asymptotic regions greatly reduced the processing time, however, also degraded the accuracy of the near-field solution beyond acceptable limits.
6. The computer execution times were much longer than anticipated.

B. RECOMMENDATIONS AND EXTENSIONS

The groundwork for developing and testing the SET has been put in place in this research. Further investigation is required and should include the following:

1. Detailed analysis of the sampling rate requirement.
2. In-depth analysis of specific contributions to the analytic and integral portions of Equation (20).
3. Incorporate SET into the Field Feedback Formulation [Ref. 11].
4. Investigate the strong effect the offset distance has on the SET near-fields.
5. Evaluate the SET for objects with exact solutions other than the circular cylinder.
6. Modify the algorithm or computer implementation to yield faster execution times without sacrificing accuracy.
7. Investigate the relative accuracy between the various available integration routines utilized.

APPENDIX A. COORDINATE GENERATION ROUTINES

A. PROGRAM DESCRIPTION

These programs generate the Cartesian coordinates which define the contours of typically shaped objects. The routines can be used individually in the NEARFLD program to provide the node points on each contour.

These programs were written by Prof. R. Janaswamy.

B. PROGRAM LISTINGS

The following are listings of four typical routines which can be used to generate the node points required by the NEARFLD program.

1. Program CIRCLE

```
PROGRAM CIRCLE
PRINT *, 'READ IN RADIUS OF CIRCLE, # OF POINTS'
READ (5,*) A, N
OPEN (UNIT = 1, FILE = 'CIRC', FORM='FORMATTED')
PI = 4. * ATAN (1.)
DELT = 2. * PI / FLOAT (N)
THETA = 0.
DO 1 I = 1, N
X = A * COS (THETA)
Y = A * SIN (THETA)
WRITE (1,*) X, Y
THETA = THETA - DELT
1 CONTINUE
END
```

2. Program SQUARE

```
PROGRAM SQUARE
PRINT *, 'READ IN SQUARE SIDE, NPTS PER SIDE'
READ (5,*) A, NPTS
```

```

OPEN (UNIT = 1, FILE = 'SQR', STATUS = 'UNKNOWN')
B = A / SQRT (2.)
DELT = B / FLOAT (NPTS)
DO 1 I = 1, NPTS
X = (I-1) * DELT
Y = B - X
1 WRITE (1,*) X, Y
CONTINUE
DO 2 I = 1, NPTS
X = B - (I-1) * DELT
Y = X - B
2 WRITE (1,*) X, Y
CONTINUE
DO 3 I = 1, NPTS
X = -(I-1) * DELT
Y = - (B + X)
3 WRITE (1,*) X, Y
CONTINUE
DO 4 I = 1, NPTS
X = -(B - (I-1) * DELT)
Y = B + X
4 WRITE (1,*) X, Y
CONTINUE
END

```

3. Program SHELL

```

PROGRAM SHELL
PRINT *, 'READ inner rad, no of pts, outer rad, no of pts, npts'
READ (5,*) A, N1, B, N2, N
OPEN (UNIT = 1, FILE = 'SHELL', FORM='FORMATTED')
PI = 4. * ATAN (1.)
DELT1 = PI / FLOAT (N1)
DELT2 = PI / FLOAT (N2)
DELT3 = (B-A) / FLOAT (2 * N)
X = 0.
Y = (A + B) / 2.
DO 4 I = 1, N + 1
WRITE (1, *) X, Y
Y = Y + DELT3
4 CONTINUE
THETA = PI / 2. - DELT2
DO 1 I = 1, N2
X = B * COS (THETA)
Y = B * SIN (THETA)
WRITE (1,*) X, Y
THETA = THETA - DELT2
1 CONTINUE
DO 2 I = 1, 2 * N

```

```

      Y = Y + DELT3
      WRITE (1, *) X, Y
2     CONTINUE
      THETA = -PI / 2.
      DO 3 I = 1, N2
      THETA = THETA + DELT2
      X = A * COS (THETA)
      Y = A * SIN (THETA)
      WRITE (1,*) X, Y
3     CONTINUE
      X = 0.
      DO 5 I = 1, N-1
      Y = Y + DELT3
      WRITE (1,*) X, Y
5     CONTINUE
      END

```

4. Program SLAB

```

PROGRAM SLAB
REAL L, T
PRINT *, 'READ LENGTH, # OF SEGS ALONG LENGTH'
READ (5,*) L, N1
PRINT *, 'READ THICKNESS, # OF SEGS ALONG WIDTH (EVEN)'
READ (5,*) T, N2
OPEN (UNIT = 1, FILE = 'SLAB', STATUS = 'UNKNOWN')
X = 0
Y = L / 2.
WRITE (1,*) X, Y
N3 = N2 / 2
DELT2 = T / FLOAT (N2)
DO 1 I = 1, N3
X = I * DELT2
WRITE (1,*) X, Y
1  CONTINUE
DELT1 = L / FLOAT (N1)
DO 2 I = 1, N1
Y = L / 2. - I * DELT1
WRITE (1,*) X, Y
2  CONTINUE
DO 3 I = 1, N2
X = T / 2. - I * DELT2
WRITE (1,*) X, Y
3  CONTINUE
DO 4 I = 1, N1
Y = - L / 2. + I * DELT1
WRITE (1,*) X, Y
4  CONTINUE
DO 5 I = 1, N3 - 1

```

```
5      X = - T / 2. + I * DELT2  
      WRITE (1,*) X, Y  
      CONTINUE  
      END
```

APPENDIX B. INFINITE SERIES FIELD SOLUTIONS

The incident and scattered fields ($\psi^{(i)}$ and $\psi^{(s)}$) for a uniform plane wave traveling in the +x direction in free space, incident normally on a lossless dielectric circular cylinder of radius a , can be found from the infinite series solutions that follow:

$$\psi^{(i)} = \hat{a}_z \psi_0 \sum_{n=-\infty}^{+\infty} j^{-n} J_n(k_0 \rho) e^{jn\phi}, \quad (\text{B-1})$$

and

$$\psi^{(s)} = \hat{a}_z \psi_0 \sum_{n=-\infty}^{+\infty} a_n H_n^{(2)}(k_0 \rho) e^{jn\phi}, \quad (\text{B-2})$$

where

$$a_n = j^{-n} \frac{J_n'(k_0 a) J_n(k_1 a) - \sqrt{\alpha \beta} J_n(k_0 a) J_n'(k_1 a)}{\sqrt{\alpha \beta} J_n'(k_1 a) H_n^{(2)}(k_0 a) - J_n(k_1 a) H_n^{(2)'}(k_0 a)} \quad (\text{B-3})$$

J_n and H_n are Bessel and Hankel functions of order n , respectively, with normal derivatives J' and H' , k_0 is the free-space wavenumber, and k_1 is the wavenumber in the dielectric [Ref. 3]. For the TM case, $\alpha = 1/\mu_r$ and $\beta = \epsilon_r$ whereas, for the TE case, $\alpha = 1/\epsilon_r$ and $\beta = \mu_r$.

The normal derivatives of the field solutions ($\psi^{(i)'}$ and $\psi^{(s)'}$) can be found from the following [Ref. 3]:

$$\psi^{(n)'} = \hat{a}_z k_0 \psi_0 \sum_{n=-\infty}^{+\infty} j^{-n} J_n'(k_0 \rho) e^{jn\phi}, \quad (\text{B-4})$$

$$\psi^{(s)'} = \hat{a}_z k_0 \psi_0 \sum_{n=-\infty}^{+\infty} a_n J_n'(k_0 \rho) e^{jn\phi}. \quad (\text{B-5})$$

APPENDIX C. NEARFLD PROGRAM

A. PROGRAM DESCRIPTION

This program prepares the input data for the SET program. It calculates the required input data and stores it in corresponding matrices. The program asks for certain quantities to be specified by the user, such as radius, frequency, nodes, etc. The parameters are outlined in the description block of the program. The program as it appears here is set up for circular cylindrical geometry. It can, however, be adapted to another geometry by replacing the subroutine CIRCLE with a suitable coordinate generation program, such as those in Appendix A.

This program was written by Lt. R. A. Rostant except where previously noted.

B. PROGRAM LISTING

PROGRAM NEARFLD

```
C
C  Program to calculate the scattered field at each of the field points (Qi)
C  utilizing the SET subroutine. This program reads the input parameters
C  and calculates the input parameters required by the SET routine.
C
C  Written by Lt. R. A. Rostant.
C
C  Input Parameters:
C      A      - Radius of cylinder in meters
C      F0     - Frequency of the incident plane wave in Hertz
C      PERND  - Number of nodes on the perimeter contour
C      LOOPS  - Number of iterations of the trapezoid rule
C              [2^(LOOPS-1)]
C      FAC   - Factor used to determine the upper limit of summation
C              the series solutions (1.5 to 2.0 is generally sufficient)
C      ER    - Relative permittivity
C      MR    - Relative permeability
C      DELTA - One-half the length of contour C2 in meters
```

```

C      OFFSET - Offset distance, i.e. normal distance between the
C      perimeter and boundary contours
C      EPS1  - Factor used to determine if the asymptotic regions of
C      contour C1 are to be considered in the SET solution
C
C      Output:
C      FILE1 - Values of the scattered field on the boundary contour
C      as calculated by the series solution
C      FILE2 - Values of the scattered field on the perimeter contour
C      as calculated by the series solution
C      FILE3 - Values of the scattered field on the boundary contour
C      as calculated by the SET
C      FILE4 - Values of the analytic (SIMP7) and integral (SIMPT)
C      terms of the SET
C
C      INTEGER PERND,K,LOOPS
C      REAL XYSR(365,4),A,FAC,ER,MR,DELTA,PENDS(365,4)
C      REAL PNODC2(365,2),OFFSET,K0A,K0,EPS1,C,F0,LAMBDA,PI
C      REAL K0R,RHO
C      REAL*8 DA,DR
C      COMPLEX PSI(365,2),NEWPSI(365,4),PSIC2(365,2),SIMP
C      COMPLEX SPSI(365),SDPSI(365)
C
C      COMMON/NODAL/XYSR,PNODC2,PSI,PSIC2
C      CHARACTER*16 FILE1,FILE2,FILE3,FILE4
C
C      WRITE(*,*) 'ENTER RADUIS-A (IN METERS)'
C      READ(*,*) A
C      WRITE(*,*) 'ENTER FREQUENCY (Hz)'
C      READ(*,*) F0
C      WRITE(*,*) 'ENTER # OF NODES-PERND(INTEGER)'
C      READ(*,*) PERND
C      WRITE(*,*) 'ENTER N FOR 2 ^ N-1 ITERATIONS OF TRAPEZOID RULE'
C      READ(*,*) LOOPS
C      WRITE(*,*) 'ENTER FACTOR-FAC (REAL)'
C      READ(*,*) FAC
C      WRITE(*,*) 'ENTER EPSILON R-ER (REAL)'
C      READ(*,*) ER
C      WRITE(*,*) 'ENTER MU R-MR (REAL)'
C      READ(*,*) MR
C      WRITE(*,*) 'ENTER DELTA (METERS)'
C      READ(*,*) DELTA
C      WRITE(*,*) 'ENTER OFFSET (METERS)'
C      READ(*,*) OFFSET
C      WRITE(*,*) 'ENTER EPSILON 1 (METERS)'
C      READ(*,*) EPS1
C      WRITE(*,*) 'ENTER EXACT BOUNDARY PSI FILE NAME IN QUOTES'
C      READ(*,*) FILE1
C      WRITE(*,*) 'ENTER EXACT SCATTERED PERIM PSI FILE NAME IN QUOTES'
C      READ(*,*) FILE2

```

```

WRITE(*,*) 'ENTER CALCULATED BOUNDARY PSI FILE NAME IN QUOTES'
READ(*,*) FILE3
WRITE(*,*) 'ENTER SIMP7/SIMPT FILE NAME IN QUOTES'
READ(*,*) FILE4
C
OPEN(UNIT=1,FILE=FILE1,STATUS='UNKNOWN')
OPEN(UNIT=2,FILE=FILE2,STATUS='UNKNOWN')
OPEN(UNIT=3,FILE=FILE3,STATUS='UNKNOWN')
OPEN(UNIT=4,FILE=FILE4,STATUS='UNKNOWN')
C
PI=4.0*ATAN(1.0)
C=2.997925E+08
LAMBDA=C/F0
K0=2*PI/LAMBDA
R=A+OFFSET
K0A=A*K0
K0R=R*K0
DA=DBLE(K0A)
DR=DBLE(K0R)
C
C Calculate node coordinates on perimeter and boundary contours
C
CALL CIRCLE(A,PERND,OFFSET,XYSR)
C
C Calculate scattered field on boundary contour using exact solution
C
CALL SCAT(FAC,ER,MR,PERND,DA,DR,SPSI)
DO 20 J=1,PERND
  WRITE(1,*) CABS(SPSI(J))
20 CONTINUE
C
C Calculate scattered field on perimeter contour using exact solution
C
CALL SCAT(FAC,ER,MR,PERND,DA,DA,SPSI)
DO 30 J=1,PERND
  PSI(J,1)=SPSI(J)
  WRITE(2,*) CABS(SPSI(J)),SPSI(J)
30 CONTINUE
C
C Calculte normal derivative of scattered field on perimeter contour
C using exact solution
C
CALL DSCAT(FAC,ER,MR,PERND,DA,K0,SDPSI)
DO 40 J=1,PERND
  PSI(J,2)=SDPSI(J)
40 CONTINUE
C
C Calculate endnodes of contour C1k for each node k
C
CALL ENDNODES(XYSR,DELTA,PERND,PENDS)

```

```

C
C Calculate  $\psi$  and  $d\psi/dn$  corresponding to each endnode generated
C by ENDNODES subroutine
C
  CALL NODEPSI(XYSR,PSI,DELTA,PERND,NEWPSI)
C
C Calculate  $\psi$  on boundary contour for each point (node) Qk
C
  DO 50 K=1,PERND
    WRITE(*,*) 'Calculating scattered field at node ',k
C
C Reorder the coordinates to reflect the proper order of the nodes
C corresponding to the k'th contour C1k
C
    CALL REORD(XYSR,PENDS,PERND,K,PNODC2)
C
C Reorder the values of  $\psi$  and  $d\psi/dn$  to correspond to the reordered
C nodes
C
    CALL CREORD(PSI,NEWPSI,PERND,K,PSIC2)
C
C Calculate the scattered field at the k'th field point Qk
C
    CALL SET(LOOPS,K0,EPS1,PERND,K,DELTA,SIMP)
    WRITE(3,*) CABS(SIMP)
50 CONTINUE
  STOP
  END

SUBROUTINE CIRCLE(K0A,N,OFFSET,XYSR)
  REAL XYSR(365,4),K0A,XK0A
  XK0A=K0A
  PI=4. * ATAN(1.)
  DTR=PI/180.
  STEP=360.0/FLOAT(N)
  K=1
  DO 2 J=1,2
    M=1
    DO 1 S=360.,STEP,-STEP
      THETA=DTR*S
      X=XK0A*COS(THETA)
      Y=XK0A*SIN(THETA)
      XYSR(M,K)=X
      XYSR(M,K+1)=Y
      M=M+1
1    CONTINUE
      K=3
      XK0A=XK0A+OFFSET
2    CONTINUE

```

RETURN
END

SUBROUTINE SCAT(FAC,ER,MR,NODES,DK0A,DK0R,MPSI)

C

C Computing 2-D Dielectric Cylinder ψ values

C

INTEGER NODES,NPHI,NMX,NMAX,N
REAL*8 J(0:365),J1(0:365),Y(0:365),Y1(0:365),DJ(0:365),
*DJ1(0:365),DY(0:365),DY1(0:365),JR(0:365),YR(0:365),DJR(0:365),
*DYR(0:365),DK0A,DK1A,DK0R
REAL K0,K1,ER,MR,STEP,PI,DTR,A,PHI1,PHI2,M
COMPLEX TPSI,PSI,MPSI(365)

C

PI=4.0*ATAN(1.0)
DTR=PI/180.
DK1A=SQRT(ER*MR)*DK0A
NMX=INT(FAC*DK0A)+1
NMAX=NMX+1
CALL BES(NMAX,DK0A,J,Y,DJ,DY)
CALL BES(NMAX,DK1A,J1,Y1,DJ1,DY1)
CALL BES(NMAX,DK0R,JR,YR,DJR,DYR)
NPHI=NODES+1
STEP=360.0/(NPHI-1.)
L=1

C

*** Stepping Through Phi = 360 to 0 deg
DO 33 M=360.,STEP,-STEP
PHI=DTR*M

C

*** Initializing Coefficients
PSI=(DCMPLX(JR(0),-YR(0)))*((DJ(0)*J1(0))-(SQRT(ER/MR)*J(0)*
* DJ1(0)))/(SQRT(ER/MR)*DJ1(0)*DCMPLX(J(0),-Y(0))-J1(0)*
* DCMPLX(DJ(0),-DY(0)))

C

*** Summing Fields
DO 22 N=1,NMX
TPSI=COS(N*PHI)*(DCMPLX(JR(N),-YR(N)))*1/((0.,1.)**N)*
* ((DJ(N)*J1(N))-(SQRT(ER/MR)*J(N)*DJ1(N)))/(SQRT(ER/MR)*DJ1(N)*
* DCMPLX(J(N),-Y(N))-J1(N)*DCMPLX(DJ(N),-DY(N)))
PSI=PSI+2.0*TPSI

22 CONTINUE

MPSI(L)=PSI

L=L+1

33 CONTINUE

RETURN

END

SUBROUTINE DSCAT(FAC,ER,MR,NODES,DK0A,K0,MPSI)

C

C Computing 2-D Dielectric Cylinder scattered $d\psi/dn$ values

```

C
INTEGER NODES,NPHI,NMX,NMAX,N
REAL*8 J(0:365),J1(0:365),Y(0:365),Y1(0:365),DJ(0:365),
*DJ1(0:365),DY(0:365),DY1(0:365),JR(0:365),YR(0:365),DJR(0:365),
*DYR(0:365),DK0A,DK1A
REAL K0,K1,ER,MR,STEP,PI,DTR,A,PHI1,PHI2,M
COMPLEX TPSI,PSI,MPSI(365)

C
PI=4.0*ATAN(1.0)
DTR=PI/180.
DK1A=SQRT(ER*MR)*DK0A
NMX=INT(FAC*DK0A)+1
NMAX=NMX+1
CALL BES(NMAX,DK0A,J,Y,DJ,DY)
CALL BES(NMAX,DK1A,J1,Y1,DJ1,DY1)
NPHI=NODES+1
STEP=360./(NPHI-1.)
L=1

C   *** Stepping Through Phi = 360 to 0 deg
DO 33 M=360.,STEP,-STEP
PHI=DTR*M
C   *** Initializing Coefficients
PSI=K0*(DCMPLX(DJ(0),-DY(0)))*((DJ(0)*J1(0))-(SQRT(ER/MR)*J(0)*
* DJ1(0)))/(SQRT(ER/MR)*DJ1(0)*DCMPLX(J(0),-Y(0))-J1(0)*
* DCMPLX(DJ(0),-DY(0)))
C   *** Summing Fields
DO 22 N=1,NMX
TPSI=K0*COS(N*PHI)*(DCMPLX(DJ(N),-DY(N)))*1/((0.,1.)**N)*
* ((DJ(N)*J1(N))-(SQRT(ER/MR)*J(N)*DJ1(N)))/(SQRT(ER/MR)*DJ1(N)*
* DCMPLX(J(N),-Y(N))-J1(N)*DCMPLX(DJ(N),-DY(N)))
PSI=PSI+2.0*TPSI
22 CONTINUE
MPSI(L)=PSI
L=L+1
33 CONTINUE
RETURN
END

SUBROUTINE INCID(FAC,NODES,DK0R,MPSI)
C
C   Computing 2-D Dielectric Cylinder incident  $\psi$  values
C
INTEGER NODES,NPHI,NMX,NMAX,N
REAL*8 J(0:365),J1(0:365),Y(0:365),Y1(0:365),DJ(0:365),
*DJ1(0:365),DY(0:365),DY1(0:365),DK0R,R1
REAL K0,K1,STEP,PI,DTR,A,PHI1,PHI2,M
COMPLEX TPSI,PSI,MPSI(365)
C

```

```

PI=4.0*ATAN(1.0)
DTR=PI/180.
NMX=INT(FAC*DK0R)+1
NMAX=NMX+1
CALL BES(NMAX,DK0R,J,Y,DJ,DY)
NPHI=NODES+1
STEP=360./(NPHI-1.)
L=1
C   *** Stepping Through Phi = 360 to 0 deg
DO 33 M=360.,STEP,-STEP
  PHI=DTR*M
C   *** Initializing Coefficients
  PSI=J(0)
C   *** Summing Fields
  DO 22 N=1,NMX
    TPSI=(COS(N*PHI)*J(N))/((0.,1.)**N)
    PSI=PSI+2.0*TPSI
  22  CONTINUE
  MPSI(L)=PSI
  L=L+1
  33  CONTINUE
  RETURN
  END

SUBROUTINE DINCID(FAC,NODES,DK0R,K0,MPSI)
C
C   Computing 2-D Dielectric Cylinder incident  $d\psi/dn$  values
C
  INTEGER NODES,NPHI,NMX,NMAX,N
  REAL*8 J(0:365),J1(0:365),Y(0:365),Y1(0:365),DJ(0:365),
  *DJ1(0:365),DY(0:365),DY1(0:365),DK0R,R1
  REAL K0,K1,STEP,PI,DTR,A,PHI1,PHI2,M
  COMPLEX TPSI,PSI,MPSI(365)
C
  PI=4.0*ATAN(1.0)
  DTR=PI/180.
  NMX=INT(FAC*DK0R)+1
  NMAX=NMX+1
  CALL BES(NMAX,DK0R,J,Y,DJ,DY)
  NPHI=NODES+1
  STEP=360./(NPHI-1.)
  L=1
C   *** Stepping Through Phi = 360 to 0 deg
DO 33 M=360.,STEP,-STEP
  PHI=DTR*M
C   *** Initializing Coefficients
  PSI=K0*DJ(0)
C   *** Summing Fields
  DO 22 N=1,NMX

```

```

        TPSI=(COS(N*PHI)*K0*DJ(N))/((0.,1.)**N)
        PSI=PSI+2.0*TPSI
22  CONTINUE
        MPSI(L)=PSI
        L=L+1
33  CONTINUE
        RETURN
        END

```

```

SUBROUTINE BES(N,X,J,Y,DJ,DY)
C
C Double precision calculation of ordinary Bessel functions,  $J_n(X)$ 
C and  $Y_n(X)$ , and their first derivative, DJ and DY, for integer
C order "n" from n=0 to N with real argument X.
C
REAL*8 J(0:365),Y(0:365),DJ(0:365),DY(0:365),SCALE,JTEMP2,X
REAL*8 SCLFAC,A,B,C,D,E,F,PI,JTEMP,JTEMP1
PI=3.14159265359D0
IF (X.EQ.0.0D00) THEN
C  X = 0.0 BOUNDARY CASE
        IF (N.EQ.1) THEN
                J(1) = 0.0D00
                DJ(1) = 0.5D00
        ELSE
                DO 5, I = N, 2, -1
                        J(I) = 0.0D00
                        DJ(I) = 0.0D00
5          CONTINUE
                J(1) = 0.0D00
                DJ(1) = 0.5D00
                ENDIF
                J(0) = 1.0D00
                DJ(0) = 0.0D00
                Y(N) = -1.0D-300
                DY(N) = 1.0D300
        ELSEIF (N.EQ.0) THEN
C          POLYNOMIAL EXPANSION ONLY FOR N = 0
                CALL BES0(X,J,Y,PI,DJ,DY)
        ELSE
C          RECURSION FOR ALL OTHER CASES
C          Y IS A FORWARD RECURSION
                CALL BES0(X,J,Y,PI,DJ,DY)
                Y(1) = -DY(0)
                DY(1)=Y(0) - Y(1)/X
                IF (N.EQ.1) GO TO 20
                DO 10, I = 0, N-2
                        Y(I+2) = (2.0D00*(I+1)/X)*Y(I+1) - Y(I)
                        DY(I+2) = Y(I+1) - ((I+2)/X)*Y(I+2)
10         CONTINUE

```

```

C   J IS A REVERSE RECURSION BASED ON A PAIR OF BESSEL FUNCTION
C   POINTS DERIVED FROM A TRUNCATED POWER SERIES EXPANSION. THE
C   RECURSION IS THEN SCALED TO A KNOWN VALUE, J1(X).
20  SCALE = -DJ(0)
    NSAVE = N
    IF (X.LE.N) THEN
        N = 5*N+50
        GOTO 25
    ENDIF
    N = IDNINT(N + X*X + 0.5D00)
C
25  A = 1.0D00/DFLOAT(N+1)
    B = 1.0D00/DFLOAT(N+2)
    C = 1.0D00/DFLOAT(N+3)
    D = 1.0D00/DFLOAT(N+4)
    E = 1.0D00/DFLOAT(N+5)
    F = X/2.0D00
C
    JTEMP = 1-A**2+0.5D00*A*B**4-(1.0D00/6.0D00)*A*B*C**6+
    + (1.0D00/24.0D00)*A*B*C*D**8-(1.0D00/120.0D00)*A*B*C*D**10
C
    N = N - 1
    A = 1.0D00/DFLOAT(N+1)
    B = 1.0D00/DFLOAT(N+2)
    C = 1.0D00/DFLOAT(N+3)
    D = 1.0D00/DFLOAT(N+4)
    E = 1.0D00/DFLOAT(N+5)
    F = X/2.0D00
C
    JTEMP1 = 1-A**2+0.5D00*A*B**4-(1.0D00/6.0D00)*A*B*C**6+
    + (1.0D00/24.0D00)*A*B*C*D**8-(1.0D00/120.0D00)*A*B*C*D**10
C
    DO 30, I = N+1,2,-1
        JTEMP2 = 2*((I - 1)/X)*JTEMP1 - JTEMP
        IF(DABS(JTEMP2).GE.1.0D250) THEN
            JTEMP2 = JTEMP2*1.0D-250
            JTEMP1 = JTEMP1*1.0D-250
        ENDIF
        JTEMP = JTEMP1
        JTEMP1 = JTEMP2
        IF ((I-2).LE.NSAVE) THEN
            J(I-2) = JTEMP2
        ENDIF
30  CONTINUE
C   SCALING
    N = NSAVE
    SCLFAC = SCALE/J(1)
    DO 40, I = 0, N
        J(I) = SCLFAC*J(I)
        IF (I.EQ.0) THEN

```

```

      GOTO 40
ENDIF
IF (ABS(J(I)/J(I-1)).LT.1.0D-50) THEN
      J(I) = J(I)*1.0D250
ELSEIF (ABS(J(I)/J(I-1)).GT.1.0D50) THEN
      J(I) = J(I)*1.0D-250
ENDIF
DJ(I) = J(I-1) - (I/X)*J(I)
40  CONTINUE
ENDIF
RETURN
END

C
SUBROUTINE BES0(X,J,Y,PI,DJ,DY)
C  FOR ZERO ORDER BESSEL FUNCTIONS ONLY
DIMENSION J(0:200), Y(0:200), DJ(0:200), DY(0:200)
DOUBLE PRECISION J, Y, X, PI, DJ, DY, F0, F1, THETA0
DOUBLE PRECISION THETA1, A
IF (X.LE.3.0D00) THEN
  A = X/3.0D00
  J(0) = 1.0D00 - 2.2499997D00*(A**2) + 1.2656208D00*(A**4) -
+0.3163866D00*(A**6) + 0.0444479D00*(A**8) - 0.0039444D00*(A**10) +
+0.00021D00*(A**12)
  Y(0) = (2.0D00/PI)*DLOG(X/2.0D00)*J(0) + 0.36746691D00 +
+.60559366D00*(A**2) - 0.74350384D00*(A**4) + 0.25300117D00*(A**6)
+ - 0.04261214D00*(A**8) + 0.00427916D00*(A**10) - 0.00024846D00*
+ (A**12)
  DJ(0) = -X*(.5D00-0.56249985D00*(A**2)+0.21093573D00
+*(A**4)- 0.03954289D00*(A**6) + 0.00443319D00*(A**8) - 0.00031761
+D00*(A**10) + 0.00001109D00*(A**12))
  DY(0) = (-1.0D00/X)*((2.0D00/PI)*X*DLOG(X/2D00)*(-1.0D00*
+DJ(0))-0.6366198D00+0.2212091D00*(A**2)+2.1682709D00*(A**4) -
+1.3164827D00*(A**6) + 0.3123951D00*(A**8) - 0.0400976D00*(A**10)
+ + 0.0027873D00*(A**12))
ELSE
  A = 3.0D00/X
  F0 = .79788456D00 - 0.00000077D00*A - 0.00552740D00*(A**2)
+ -0.00009512D00*(A**3) + 0.00137237D00*(A**4) -0.00072805D00*(A**5)
+ +0.00014476D00*(A**6)
  THETA0 = X - 0.78539816D00 - 0.04166397D00*A - 0.00003954
+D00*(A**2) + 0.00262573D00*(A**3) - 0.00054125D00*(A**4) -
+0.00029333D00*(A**5) + 0.00013558D00*(A**6)
  J(0) = F0*DCOS(THETA0)/DSQRT(X)
  Y(0) = F0*DSIN(THETA0)/DSQRT(X)
  F1 = 0.79788456D00 + 0.00000156D00*A + 0.01659667D00*A*A
+ +0.00017105D00*(A**3) - 0.00249511D00*(A**4) + 0.00113653D00
+*(A**5) -0.00020033D00*(A**6)
  THETA1 = X - 2.35619449D00 + .12499612D00*A + 0.00005650
+D00*(A**2) - 0.00637879D00*(A**3) + 0.00074348D00*(A**4) +
+ 0.00079824D00*(A**5) - 0.00029166D00*(A**6)

```

```

      DJ(0) = -F1*DCOS(THETA1)/DSQRT(X)
      DY(0) = -F1*DSIN(THETA1)/DSQRT(X)
ENDIF
RETURN
END

```

```

SUBROUTINE ENDNODES(MESH,DELTA,N,PENDS)
C   This subroutine computes the new end nodes (x*,y*) and
C   (x',y') for each original node on the boundary contour.
C   (x*,y*) is the first node in the clockwise direction, a
C   distance of  $\delta$  away from the corresponding k'th node.
C   (x',y') is the last node on the contour C1. The input matrix
C   'MESH' contains (xk,yk,sk,rk) and the output matrix 'PENDS'
C   contains (x*,y*,x',y').
C
REAL MESH(365,4),PENDS(365,4),DELTA,M1,X1,Y1,X2,Y2
REAL ADDER
INTEGER K
C
DO 30 K=1,N
C
  IF(ABS(MESH(K,1)-MESH(K,3)).LT.0.001) THEN
    X1=MESH(K,1)+DELTA
    X2=MESH(K,1)-DELTA
    Y1=MESH(K,2)
    S1=X1
    S2=X2
    R1=MESH(K,4)
    GO TO 20
  ENDIF
C
  M1=(MESH(K,4)-MESH(K,2))/(MESH(K,3)-MESH(K,1))
C
  IF(ABS(M1).LT.0.001) THEN
    X1=MESH(K,1)
    Y1=MESH(K,2)+DELTA
    Y2=MESH(K,2)-DELTA
    S1=MESH(K,3)
    R1=Y1
    R2=Y2
    GO TO 10
  ENDIF
C
  ADDER=DELTA*M1/SQRT(1+M1**2)
  X1=MESH(K,1)+ADDER
  X2=MESH(K,1)-ADDER
  Y1=MESH(K,2)-(X1-MESH(K,1))/M1
  Y2=MESH(K,2)-(X2-MESH(K,1))/M1
  S1=MESH(K,3)+ADDER

```

```

S2=MESH(K,3)-ADDER
R1=MESH(K,4)-(S1-MESH(K,3))/M1
R2=MESH(K,4)-(S2-MESH(K,3))/M1
C
IF((MESH(K,3).GT.MESH(K,1)).AND.(MESH(K,4).GT.MESH(K,2))) THEN
  PENDS(K,1)=AMAX1(X1,X2)
  PENDS(K,2)=AMIN1(Y1,Y2)
  PENDS(K,3)=AMIN1(X1,X2)
  PENDS(K,4)=AMAX1(Y1,Y2)
  GO TO 30
ELSEIF((MESH(K,3).LT.MESH(K,1)).AND.(MESH(K,4).LT.MESH(K,2)))
* THEN
  PENDS(K,1)=AMIN1(X1,X2)
  PENDS(K,2)=AMAX1(Y1,Y2)
  PENDS(K,3)=AMAX1(X1,X2)
  PENDS(K,4)=AMIN1(Y1,Y2)
  GO TO 30
ELSEIF((MESH(K,3).GT.MESH(K,1)).AND.(MESH(K,4).LT.MESH(K,2)))
* THEN
  PENDS(K,1)=AMIN1(X1,X2)
  PENDS(K,2)=AMIN1(Y1,Y2)
  PENDS(K,3)=AMAX1(X1,X2)
  PENDS(K,4)=AMAX1(Y1,Y2)
  GO TO 30
ELSE
  PENDS(K,1)=AMAX1(X1,X2)
  PENDS(K,2)=AMAX1(Y1,Y2)
  PENDS(K,3)=AMIN1(X1,X2)
  PENDS(K,4)=AMIN1(Y1,Y2)
  GO TO 30
ENDIF
C
10 IF(MESH(K,3).GT.MESH(K,1)) THEN
  PENDS(K,1)=X1
  PENDS(K,2)=AMIN1(Y1,Y2)
  PENDS(K,3)=X1
  PENDS(K,4)=AMAX1(Y1,Y2)
  GO TO 30
ELSE
  PENDS(K,1)=X1
  PENDS(K,2)=AMAX1(Y1,Y2)
  PENDS(K,3)=X1
  PENDS(K,4)=AMIN1(Y1,Y2)
  GO TO 30
ENDIF
C
20 IF(MESH(K,4).GT.MESH(K,2)) THEN
  PENDS(K,1)=AMAX1(X1,X2)
  PENDS(K,2)=Y1
  PENDS(K,3)=AMIN1(X1,X2)

```

```

      PENDS(K,4)=Y1
      GO TO 30
    ELSE
      PENDS(K,1)=AMIN1(X1,X2)
      PENDS(K,2)=Y1
      PENDS(K,3)=AMAX1(X1,X2)
      PENDS(K,4)=Y1
    ENDIF
  C
30 CONTINUE
  RETURN
  END

SUBROUTINE NODEPSI(XYSR,PSI,DELTA,N,NEWPSI)
  C
  C   Subroutine to calculate values of  $\psi$  and  $d\psi/dn$  at the new
  C   endnodes for each node k.
  C
  REAL DELTA,LMIN,LPLUS,XYSR(365,4)
  INTEGER K,N
  COMPLEX PSI(365,2),NEWPSI(365,4),SIK,SIMIN,SIPLUS,DSIK,DSIMIN
  COMPLEX DSIPLUS
  C
  DO 10 K=1,N
    SIK=PSI(K,1)
    DSIK=PSI(K,2)
    IF(K.EQ.1) THEN
      SIMIN=PSI(N,1)
      SIPLUS=PSI(2,1)
      DSIMIN=PSI(N,2)
      DSIPLUS=PSI(2,2)
      LMIN=SQRT((XYSR(N,1)-XYSR(1,1))**2+
      * (XYSR(N,2)-XYSR(1,2))**2)
      LPLUS=SQRT((XYSR(2,1)-XYSR(1,1))**2+
      * (XYSR(2,2)-XYSR(1,2))**2)
    ELSEIF(K.EQ.N) THEN
      SIMIN=PSI(N-1,1)
      SIPLUS=PSI(1,1)
      DSIMIN=PSI(N-1,2)
      DSIPLUS=PSI(1,2)
      LMIN=SQRT((XYSR(N-1,1)-XYSR(N,1))**2+
      * (XYSR(N-1,2)-XYSR(N,2))**2)
      LPLUS=SQRT((XYSR(1,1)-XYSR(N,1))**2+
      * (XYSR(1,2)-XYSR(N,2))**2)
    ELSE
      SIMIN=PSI(K-1,1)
      SIPLUS=PSI(K+1,1)
      DSIMIN=PSI(K-1,2)
      DSIPLUS=PSI(K+1,2)
    ENDIF
  END DO

```

```

      LMIN=SQRT((XYSR(K-1,1)-XYSR(K,1))**2+
*      (XYSR(K-1,2)-XYSR(K,2))**2)
      LPLUS=SQRT((XYSR(K+1,1)-XYSR(K,1))**2+
*      (XYSR(K+1,2)-XYSR(K,2))**2)
      ENDIF
      NEWPSI(K,1)=SIK+DELTA/LPLUS*(SIPLUS-SIK)
      NEWPSI(K,2)=DSIK+DELTA/LPLUS*(DSIPLUS-DSIK)
      NEWPSI(K,3)=SIK+DELTA/LMIN*(SIMIN-SIK)
      NEWPSI(K,4)=DSIK+DELTA/LMIN*(DSIMIN-DSIK)
10 CONTINUE
      RETURN
      END

```

```

      SUBROUTINE REORD(MESH,PENDS,N,K,PNODC2)
C
C      Subroutine to reorder the perimeter nodes from the start node
C      to the stop node on the contour C1. The input matrix 'MESH'
C      contains the (x,y) and (s,r) node points. The (x,y) nodes are
C      reordered with the new endnodes from 'PENDS' added to the
C      beginning and end of the matrix. The k'th node is deleted as
C      well and the new matrix is called 'PNODC2'.
C
      INTEGER I,J,K,N,INDEX
      REAL MESH(365,4),PENDS(365,4),PNODC2(365,2)
C
      PNODC2(1,1)=PENDS(K,1)
      PNODC2(1,2)=PENDS(K,2)
      DO 20 I=2,N-K+1
        DO 10 J=1,2
          PNODC2(I,J)=MESH(K+I-1,J)
10      CONTINUE
20      CONTINUE
C
C      AT THIS POINT WE HAVE THE FIRST N-K+1 POINTS IN THE MATRIX
C      MESH2. NOW FILL IN THE LAST K TERMS.
C
      INDEX=1
      DO 40 I=N-K+2,N
        DO 30 J=1,2
          PNODC2(I,J)=MESH(INDEX,J)
30      CONTINUE
        INDEX=INDEX+1
40      CONTINUE
      PNODC2(N+1,1)=PENDS(K,3)
      PNODC2(N+1,2)=PENDS(K,4)
      RETURN
      END

```

```

SUBROUTINE CREORD(MESH,ENDS,N,K,MESH2)
C
C Subroutine to reorder the values of  $\psi$  and  $d\psi/dn$  at each
C perimeter node from the start node to the stop node on
C contour C1.
C
  INTEGER I,J,K,N,INDEX
  COMPLEX MESH(365,4),ENDS(365,4),MESH2(365,2)
C
  MESH2(1,1)=ENDS(K,1)
  MESH2(1,2)=ENDS(K,2)
  DO 20 I=2,N-K+1
    DO 10 J=1,2
      MESH2(I,J)=MESH(K+I-1,J)
10  CONTINUE
20  CONTINUE
C
C AT THIS POINT WE HAVE THE FIRST N-K+1 POINTS IN THE MATRIX
C MESH2. NOW FILL IN THE LAST K TERMS.
C
  INDEX=1
  DO 40 I=N-K+2,N
    DO 30 J=1,2
      MESH2(I,J)=MESH(INDEX,J)
30  CONTINUE
    INDEX=INDEX+1
40  CONTINUE
  MESH2(N+1,1)=ENDS(K,3)
  MESH2(N+1,2)=ENDS(K,4)
  RETURN
  END

```

APPENDIX D. SINGULARITY EXTRACTION PROGRAM

A. PROGRAM DESCRIPTION

This subprogram calculates the scattered field on the boundary contour of an arbitrary dielectric object. The program receives the required input data from the program NEARFLD and calculates the scattered field at the specified point using the Singularity Extraction Technique developed in Chapter II. The program appears with a trapezoid rule integration routine (TRAP), however an alternate integration routine, such as SIMP or CADRE, may be substituted.

This program was written by Lt. R. A. Rostant except where previously noted.

B. PROGRAM LISTING

```
SUBROUTINE SET(LOOPS,K0,EPS1,SEG,K,DELTA,SIMP)
C  Subroutine to calculate the scattered field at the field point
C  ( $Q_j$ ) utilizing the SET. For each call to SET, the main calling
C  program must provide the required input parameters discussed in
C  program NEARFLD. SET calculates  $\psi^{(s)}(Q_j)$  by evaluating the
C  analytic and integral terms of the SET equation and summing for
C  final result. The integration may be accomplished using any valid
C  numerical integration routine. This program performs its
C  calculations strictly utilizing the coordinates input in PNODC2 and
C  the associated field quantities in PSIC2.
C
C  Arguments:
C    LOOPS - Number of iterations by trapezoid integration
C           [2^(LOOPS-1)]
C    K0    - Free-space wavenumber
C    EPS1  - Factor used to determine if the asymptotic regions
C           of contour C1 are considered in the SET solution
C    SEG   - Number of nodes on the perimeter contour
C    K     - Number of the node being considered
C    DELTA - One-half of the length of contour C2
C    SIMP  - Calculated scattered field on the boundary
```

```

C          contour ( $\psi_B^{(s)}$ )
C
EXTERNAL ARG1A,ARG1B,ARG2A,ARG2B,ARG3A,ARG3B
EXTERNAL ARG4A,ARG4B,ARG5A,ARG5B,ARG6A,ARG6B
C
INTEGER SEG,K,LL,KK,LOOPS
REAL X,Y,XK,YK,XK1,YK1,XB,YB,K0,XK0,R,COSTH,SINTH
REAL LK,SS1,SS2,PI,DELTA,Z,ATDELZ
REAL XYSR(365,4),PNODC2(365,2)
COMPLEX PSI(365,2),PSIC2(365,2)
COMPLEX SIMP,J,SIK,DSIK,SIK1,DSIK1,SIMPT,SIQ,DSIQ
COMPLEX SIMP1,SIMP2,SIMP3,SIMP4,SIMP5,SIMP6,SIMP7
C
COMMON/ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
COMMON/NODAL/XYSR,PNODC2,PSI,PSIC2
C
J=(0.,1.)
PI=4.0*ATAN(1.0)
SIMP=(0.,0.)
SIMPT=(0.,0.)
XK0=K0
C
C   THIS IS THE NODE POINT OF INTEREST AND ITS CORRESPONDING
C   NORMAL POINT
C
XB=XYSR(K,1)
YB=XYSR(K,2)
X=XYSR(K,3)
Y=XYSR(K,4)
SIQ=PSI(K,1)
DSIQ=PSI(K,2)
Z=SQRT((X-XB)**2 + (Y-YB)**2)
ATDELZ=ATAN(DELTA/Z)
C
C   CONTRIBUTION FROM CONTOUR C2
C
SIMP7=-SIQ*(ATDELZ/PI-1.5)+((DELTA*DSIQ/PI)*(Z/DELTA*
* ATDELZ+0.5*ALOG(1.0+(Z/DELTA)**2)))
C
C   CALCULATE THE CONTRIBUTION FROM EACH INTEGRAL ALONG EACH
C   SEGMENT (Sk).
C
DO 100 I=1,SEG
  XK=PNODC2(I,1)
  YK=PNODC2(I,2)
  XK1=PNODC2(I+1,1)
  YK1=PNODC2(I+1,2)
  SIK=PSIC2(I,1)
  DSIK=PSIC2(I,2)
  SIK1=PSIC2(I+1,1)

```

```

DSIK1=PSIC2(I+1,2)
LK = SQRT((XK1-XK)**2 + (YK1-YK)**2)
COSTH = (XK1-XK)/LK
SINTH = (YK1-YK)/LK
R0=SQRT((X-XK)**2 + (Y-YK)**2)
RB0=SQRT((XB-XK)**2 + (YB-YK)**2)
RLK=SQRT((X-XK-LK*COSTH)**2 + (Y-YK-LK*SINTH)**2)
RBLK=SQRT((XB-XK-LK*COSTH)**2 + (YB-YK-LK*SINTH)**2)
DIF1=XK0*(R0-RB0)
DIF2=XK0*(RLK-RBLK)
C
C ARE THE R AND RBAR VECTORS AT K AND K+1 EQUIVALENT IN LENGTH.
C IF SO, THERE IS NO CONTRIBUTION.
C
IF (ABS(DIF1) .LT. EPS1 .AND. ABS(DIF2) .LT. EPS1) THEN
  SIMPT=(0.,0.)
ELSE
C
C CALCULATE INTEGRAL 1
C
LL=LOOPS
DO 51 KK=1,LL+1
  CALL TRAP(ARG1A,LK,SS1,KK)
51 CONTINUE
LL=LOOPS
DO 52 KK=1,LL+1
  CALL TRAP(ARG1B,LK,SS2,KK)
52 CONTINUE
SIMP1=(DSIK/(4*J))*(SS1-J*SS2)
C
C CALCULATE INTEGRAL 2
C
LL=LOOPS
DO 53 KK=1,LL+1
  CALL TRAP(ARG2A,LK,SS1,KK)
53 CONTINUE
LL=LOOPS
DO 54 KK=1,LL+1
  CALL TRAP(ARG2B,LK,SS2,KK)
54 CONTINUE
SIMP2=((DSIK1-DSIK)/(4*J*LK))*(SS1-J*SS2)
C
C CALCULATE INTEGRAL 3
C
LL=LOOPS
DO 55 KK=1,LL+1
  CALL TRAP(ARG3A,LK,SS1,KK)
55 CONTINUE
LL=LOOPS
DO 56 KK=1,LL+1

```

```

        CALL TRAP(ARG3B,LK,SS2,KK)
56  CONTINUE
    SIMP3=((XK0*SIK*SINTH)/(4*J))*(SS1-J*SS2)
C
C    CALCULATE INTEGRAL 4
C
    LL=LOOPS
    DO 57 KK=1,LL+1
        CALL TRAP(ARG4A,LK,SS1,KK)
57  CONTINUE
    LL=LOOPS
    DO 58 KK=1,LL+1
        CALL TRAP(ARG4B,LK,SS2,KK)
58  CONTINUE
    SIMP4=((XK0*SIK*COSTH)/(4*J))*(SS1-J*SS2)
C
C    CALCULATE INTEGRAL 5
C
    LL=LOOPS
    DO 59 KK=1,LL+1
        CALL TRAP(ARG5A,LK,SS1,KK)
59  CONTINUE
    LL=LOOPS
    DO 60 KK=1,LL+1
        CALL TRAP(ARG5B,LK,SS2,KK)
60  CONTINUE
    SIMP5=((SIK1-SIK)*SINTH*XK0)/(4*J*LK)*(SS1-J*SS2)
C
C    CALCULATE INTEGRAL 6
C
    LL=LOOPS
    DO 61 KK=1,LL+1
        CALL TRAP(ARG6A,LK,SS1,KK)
61  CONTINUE
    LL=LOOPS
    DO 62 KK=1,LL+1
        CALL TRAP(ARG6B,LK,SS2,KK)
62  CONTINUE
    SIMP6=((SIK1-SIK)*COSTH*XK0)/(4*J*LK)*(SS1-J*SS2)
C
    SIMPT=SIMPT-SIMP1-SIMP2-SIMP3+SIMP4-SIMP5+SIMP6
    ENDIF
100 CONTINUE
    WRITE(4,110) SIMP7,SIMPT,CABS(SIMP7),CABS(SIMPT)
    SIMP=SIMPT-SIMP7
110  FORMAT('(',f8.5,1x,f8.5,')',2x,'(',f8.5,1x,f8.5,')',2x,f8.5,1x,
     &      cf8.5)
    RETURN
    END

```

```

SUBROUTINE TRAP(FUNC,B,S,N)
C
C   Computes the N'th stage of refinement of an extended trapezoidal
C   rule. FUNC is input as the name of the function to be integrated
C   between limits 0 and B, also input. (Can be modified for limits
C   A to B.) When called with N=1, the routine returns as S the crudest
C   estimate of the integral. Subsequent call with N=2,3,... (in that
C   sequential order will improve the accuracy of S by adding  $2^{N-2}$ 
C   additional interior points. S should not be modified between
C   sequential calls. Yields  $2^{N-1}$  segments.
C

```

```

IF(N.EQ.1) THEN
  S=0.5*B*(FUNC(0.)+FUNC(B))
  IT=1
ELSE
  TNM=IT
  DEL=B/TNM
  TAU=0.5*DEL
  SUM=0.
  DO 30 J=1,IT
    SUM=SUM+FUNC(TAU)
    TAU=TAU+DEL
30  CONTINUE
  S=0.5*(S+B*SUM/TNM)
  IT=2*IT
ENDIF
RETURN
END

```

```

REAL FUNCTION ARG1A(T)
C
C   COMPUTES - ARGUMENT FOR INTEGRAL 1
C
REAL R,RB,JO,JOB
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

JO=BESSJ0(XK0*R)
JOB=BESSJ0(XK0*RB)
ARG1A=JO-JOB
RETURN
END

```

```

REAL FUNCTION ARG1B(T)

```

```

C
C   COMPUTES ARGUMENT FOR INTEGRAL 2
C
REAL R,RB,YO,YOB
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

YO=BESSY0(XK0*R)
YOB=BESSY0(XK0*RB)
ARG1B=YO-YOB
RETURN
END

REAL FUNCTION ARG2A(T)
C
C   COMPUTES ARGUMENT FOR INTEGRAL 3
C
REAL R,RB,JO,JOB
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

JO=BESSJ0(XK0*R)
JOB=BESSJ0(XK0*RB)
ARG2A=(JO-JOB)*T
RETURN
END

REAL FUNCTION ARG2B(T)
C
C   COMPUTES ARGUMENT FOR INTEGRAL 4
C
REAL R,RB,YO,YOB
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

YO=BESSY0(XK0*R)
YOB=BESSY0(XK0*RB)
ARG2B=(YO-YOB)*T
RETURN

```

END

REAL FUNCTION ARG3A(T)

C
C COMPUTES ARGUMENT FOR INTEGRAL 5
C

REAL R,RB,J1,J1B
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

COJ1=(X-XP)/R
COJ1B=(XB-XP)/RB

J1=BESSJ1(XK0*R)
J1B=BESSJ1(XK0*RB)
ARG3A=(J1*COJ1)-(J1B*COJ1B)
RETURN
END

REAL FUNCTION ARG3B(T)

C
C COMPUTES ARGUMENT FOR INTEGRAL 6
C

REAL R,RB,Y1,Y1B
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

COJ1=(X-XP)/R
COJ1B=(XB-XP)/RB

Y1=BESSY1(XK0*R)
Y1B=BESSY1(XK0*RB)
ARG3B=(Y1*COJ1)-(Y1B*COJ1B)
RETURN
END

REAL FUNCTION ARG4A(T)

C
C COMPUTES ARGUMENT FOR INTECRAL 7
C

REAL R,RB,J1,J1B

```

COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

COJ1=(Y-YP)/R
COJ1B=(YB-YP)/RB

J1=BESSJ1(XK0*R)
J1B=BESSJ1(XK0*RB)
ARG4A=(J1*COJ1)-(J1B*COJ1B)
RETURN
END

```

```

REAL FUNCTION ARG4B(T)

```

```

C
C   COMPUTES ARGUMENT FOR INTEGRAL 8
C
REAL R,RB,Y1,Y1B
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

COJ1=(Y-YP)/R
COJ1B=(YB-YP)/RB

Y1=BESSY1(XK0*R)
Y1B=BESSY1(XK0*RB)
ARG4B=(Y1*COJ1)-(Y1B*COJ1B)
RETURN
END

```

```

REAL FUNCTION ARG5A(T)

```

```

C
C   COMPUTES ARGUMENT FOR INTEGRAL 9
C
REAL R,RB,J1,J1B
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

COJ1=(X-XP)/R
COJ1B=(XB-XP)/RB

```

```

J1=BESSJ1(XK0*R)
J1B=BESSJ1(XK0*RB)
ARG5A=((J1*COJ1)-(J1B*COJ1B))*T
RETURN
END

```

```

REAL FUNCTION ARG5B(T)

```

```

C
C   COMPUTES ARGUMENT FOR INTEGRAL 10
C
REAL R,RB,Y1,Y1B
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

COJ1=(X-XP)/R
COJ1B=(XB-XP)/RB

Y1=BESSY1(XK0*R)
Y1B=BESSY1(XK0*RB)
ARG5B=((Y1*COJ1)-(Y1B*COJ1B))*T
RETURN
END

```

```

REAL FUNCTION ARG6A(T)

```

```

C
C   COMPUTES ARGUMENT FOR INTEGRAL 11
C
REAL R,RB,J1,J1B
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

COJ1=(Y-YP)/R
COJ1B=(YB-YP)/RB

J1=BESSJ1(XK0*R)
J1B=BESSJ1(XK0*RB)
ARG6A=((J1*COJ1)-(J1B*COJ1B))*T
RETURN
END

```

```

REAL FUNCTION ARG6B(T)

```

C
C COMPUTES ARGUMENT FOR INTEGRAL 12
C

```

REAL R,RB,Y1,Y1B
COMMON /ARGS/X,Y,XB,YB,COSTH,SINTH,XK0,XK,YK
XP=T*COSTH+XK
YP=T*SINTH+YK
R=SQRT((X-XP)**2+(Y-YP)**2)
RB=SQRT((XB-XP)**2+(YB-YP)**2)

COJ1=(Y-YP)/R
COJ1B=(YB-YP)/RB

Y1=BESSY1(XK0*R)
Y1B=BESSY1(XK0*RB)
ARG6B=((Y1*COJ1)-(Y1B*COJ1B))*T
RETURN
END

```

```

FUNCTION BESSJ0(X)
REAL*8 Y,P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5,R1,R2,R3,R4,R5,R6,
* S1,S2,S3,S4,S5,S6
DATA P1,P2,P3,P4,P5/1.D0,-.1098628627D-2,.2734510407D-4,
* -.2073370639D-5,.2093887211D-6/, Q1,Q2,Q3,Q4,Q5/-.1562499995D-
*1,
* .1430488765D-3,-.6911147651D-5,.7621095161D-6,-.934945152D-7/
DATA R1,R2,R3,R4,R5,R6/57568490574.D0,-13362590354.D0,651619640.7D
*0,
* -11214424.18D0,77392.33017D0,-184.9052456D0/,
* S1,S2,S3,S4,S5,S6/57568490411.D0,1029532985.D0,
* 9494680.718D0,59272.64853D0,267.8532712D0,1.D0/
IF (ABS(X).LT.8.) THEN
  Y=X**2
  BESSJ0=(R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*R6))))
* /(S1+Y*(S2+Y*(S3+Y*(S4+Y*(S5+Y*S6))))
ELSE
  AX=ABS(X)
  Z=8./AX
  Y=Z**2
  XX=AX-.785398164
  BESSJ0=SQRT(.636619772/AX)*(COS(XX)*(P1+Y*(P2+Y*(P3+Y*(P4+Y
* *P5))))-Z*SIN(XX)*(Q1+Y*(Q2+Y*(Q3+Y*(Q4+Y*Q5))))
ENDIF
RETURN
END

```

```

FUNCTION BESSY0(X)
REAL*8 Y,P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5,R1,R2,R3,R4,R5,R6,

```

```

* S1,S2,S3,S4,S5,S6
DATA P1,P2,P3,P4,P5/1.D0,-.1098628627D-2,.2734510407D-4,
* -.2073370639D-5,.2093887211D-6/, Q1,Q2,Q3,Q4,Q5/-.1562499995D-
*1,
* .1430488765D-3,-.6911147651D-5,.7621095161D-6,-.934945152D-7/
DATA R1,R2,R3,R4,R5,R6/-2957821389.D0,7062834065.D0,-512359803.6D0
*,
* 10879881.29D0,-86327.92757D0,228.4622733D0/,
* S1,S2,S3,S4,S5,S6/40076544269.D0,745249964.8D0,
* 7189466.438D0,47447.26470D0,226.1030244D0,1.D0/
IF(X.LT.8.)THEN
  Y=X**2
  BESSY0=(R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*R6)))))/(S1+Y*(S2+Y
* *(S3+Y*(S4+Y*(S5+Y*S6))))+.636619772*BESSJ0(X)*LOG(X)
ELSE
  Z=8./X
  Y=Z**2
  XX=X-.785398164
  BESSY0=SQRT(.636619772/X)*(SIN(XX)*(P1+Y*(P2+Y*(P3+Y*(P4+Y*
* P5))))+Z*COS(XX)*(Q1+Y*(Q2+Y*(Q3+Y*(Q4+Y*Q5))))
ENDIF
RETURN
END

```

```

FUNCTION BESSJ1(X)
REAL*8 Y,P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5,R1,R2,R3,R4,R5,R6,
* S1,S2,S3,S4,S5,S6
DATA R1,R2,R3,R4,R5,R6/72362614232.D0,-7895059235.D0,242396853.1D0
*,
* -2972611.439D0,15704.48260D0,-30.16036606D0/,
* S1,S2,S3,S4,S5,S6/144725228442.D0,2300535178.D0,
* 18583304.74D0,99447.43394D0,376.9991397D0,1.D0/
DATA P1,P2,P3,P4,P5/1.D0,.183105D-2,-.3516396496D-4,.2457520174D-5
*,
* -.240337019D-6/, Q1,Q2,Q3,Q4,Q5/.04687499995D0,-.2002690873D-3
*,
* .8449199096D-5,-.88228987D-6,.105787412D-6/
IF(ABS(X).LT.8.)THEN
  Y=X**2
  BESSJ1=X*(R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*R6))))
* /(S1+Y*(S2+Y*(S3+Y*(S4+Y*(S5+Y*S6))))
ELSE
  AX=ABS(X)
  Z=8./AX
  Y=Z**2
  XX=AX-2.356194491
  BESSJ1=SQRT(.636619772/AX)*(COS(XX)*(P1+Y*(P2+Y*(P3+Y*(P4+Y
* *P5))))-Z*SIN(XX)*(Q1+Y*(Q2+Y*(Q3+Y*(Q4+Y*Q5))))
* *SIGN(1.,X)

```

```

FUNCTION BESSY1(X)
REAL*8 Y,P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5,R1,R2,R3,R4,R5,R6,
* S1,S2,S3,S4,S5,S6,S7
DATA P1,P2,P3,P4,P5/1.D0,.183105D-2,-.3516396496D-4,.2457520174D-5
*,
* -.240337019D-6/, Q1,Q2,Q3,Q4,Q5/.04687499995D0,-.2002690873D-3
*,
* .8449199096D-5,-.88228987D-6,.105787412D-6/
DATA R1,R2,R3,R4,R5,R6/-.4900604943D13,.1275274390D13,-.5153438139
*D11,
* .7349264551D9,-.4237922726D7,.8511937935D4/,
* S1,S2,S3,S4,S5,S6,S7/.2499580570D14,.4244419664D12,
* .3733650367D10,.2245904002D8,.1020426050D6,.3549632885D3,1.D0/
IF(X.LT.8.)THEN
  Y=X**2
  BESSY1=X*(R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*(R6)))))))/(S1+Y*(S2+Y*
* (S3+Y*(S4+Y*(S5+Y*(S6+Y*(S7)))))))+.636619772
* *(BESSJ1(X)*LOG(X)-1./X)
ELSE
  Z=8./X
  Y=Z**2
  XX=X-2.356194491
  BESSY1=SQRT(.636619772/X)*(SIN(XX)*(P1+Y*(P2+Y*(P3+Y*(P4+Y
* *P5))))+Z*COS(XX)*(Q1+Y*(Q2+Y*(Q3+Y*(Q4+Y*Q5))))))
ENDIF
RETURN
END

```

APPENDIX E. EXPANDED FORM OF SET INTEGRAL TERM

The following 12 expressions are an expanded form of the SET integral term in Equation (20) from Chapter II.

$$- \frac{\psi'_k}{4j} \int_{u=0}^{l_k} [J_0(k_0 R) - J_0(k_0 \bar{R})] du \quad (\text{E-1})$$

$$+ \frac{\psi'_k}{4} \int_{u=0}^{l_k} [Y_0(k_0 R) - Y_0(k_0 \bar{R})] du \quad (\text{E-2})$$

$$- \left(\frac{\psi'_{k+1} - \psi'_k}{4j l_k} \right) \int_{u=0}^{l_k} [J_0(k_0 R) - J_0(k_0 \bar{R})] u du \quad (\text{E-3})$$

$$+ \left(\frac{\psi'_{k+1} - \psi'_k}{4 l_k} \right) \int_{u=0}^{l_k} [Y_0(k_0 R) - Y_0(k_0 \bar{R})] u du \quad (\text{E-4})$$

$$- \frac{\psi_k k_0 \sin \theta_k}{4j} \int_{u=0}^{l_k} \left[\frac{J_1(k_0 R)}{R} (x - x') - \frac{J_1(k_0 \bar{R})}{\bar{R}} (\bar{x} - x') \right] du \quad (\text{E-5})$$

$$+ \frac{\psi_k k_0 \sin \theta_k}{4} \int_{u=0}^{l_k} \left[\frac{Y_1(k_0 R)}{R} (x - x') - \frac{Y_1(k_0 \bar{R})}{\bar{R}} (\bar{x} - x') \right] du \quad (\text{E-6})$$

$$+ \frac{\psi_k k_0 \cos \theta_k}{4j} \int_{u=0}^{l_k} \left[\frac{J_1(k_0 R)}{R} (y - y') - \frac{J_1(k_0 \bar{R})}{\bar{R}} (\bar{y} - y') \right] du \quad (\text{E-7})$$

$$- \frac{\psi_k k_0 \cos \theta_k}{4} \int_{u=0}^{l_k} \left[\frac{Y_1(k_0 R)}{R} (y - y') - \frac{Y_1(k_0 \bar{R})}{\bar{R}} (\bar{y} - y') \right] du \quad (\text{E-8})$$

$$- \frac{(\psi_{k+1} - \psi_k) k_0 \sin \theta_k}{4j l_k} \int_{u=0}^{l_k} \left[\frac{J_1(k_0 R)}{R} (x - x') - \frac{J_1(k_0 \bar{R})}{\bar{R}} (\bar{x} - x') \right] u du \quad (\text{E-9})$$

$$+ \frac{(\psi_{k+1} - \psi_k) k_0 \sin \theta_k}{4 l_k} \int_{u=0}^{l_k} \left[\frac{Y_1(k_0 R)}{R} (x - x') - \frac{Y_1(k_0 \bar{R})}{\bar{R}} (\bar{x} - x') \right] u du \quad (\text{E-10})$$

$$+ \frac{(\psi_{k+1} - \psi_k) k_0 \cos \theta_k}{4j l_k} \int_{u=0}^{l_k} \left[\frac{J_1(k_0 R)}{R} (y - y') - \frac{J_1(k_0 \bar{R})}{\bar{R}} (\bar{y} - y') \right] u du \quad (\text{E-11})$$

$$- \frac{(\psi_{k+1} - \psi_k) k_0 \cos \theta_k}{4 l_k} \int_{u=0}^{l_k} \left[\frac{Y_1(k_0 R)}{R} (y - y') - \frac{Y_1(k_0 \bar{R})}{\bar{R}} (\bar{y} - y') \right] u du \quad (\text{E-12})$$

where J_0 and J_1 are Bessel Functions of order zero and one, respectively, Y_0 and Y_1 are Neumann Functions of order zero and one, respectively, ψ_k and ψ'_k are the scattered field and its normal derivative on the k -th segment, as in Figure 5 from Chapter III, and l_k is the length of segment S_k .

APPENDIX F. INCIDENT FIELD INTEGRATION PROGRAM

A. PROGRAM DESCRIPTION

This program calculates the scattered field from a circular cylinder by utilizing the Green's Function Integral of Equation (2) where $\psi = \psi^{(i)}$.

This program was written by Lt. R. A. Rostant. The subroutine HAN1 was written by Prof. M. A. Morgan.

B. PROGRAM LISTING

```
PROGRAM INTEST
C
C Program to calculate the scattered field from a circular cylinder
C utilizing the Green's function contour integral for  $\psi = \psi^{(i)}$ .
C
INTEGER NSEG
REAL PI,C,RA,RP,DPHIP,ARCLN,PHIP,FREQ,LAMBDA,K0,CP,R,CA,NODES
REAL PHI,DPHI
COMPLEX J,PSI,DPSI,HH0,HH1,F,FTOT,INT,RC
C
J=(0.,1.)
PI=4.0*ATAN(1.0)
C=2.997925E+08
C
OPEN(UNIT=10,FILE='INDATA',STATUS='UNKNOWN')
WRITE(*,*) 'ENTER INNER CYLINDER RADIUS (kr*rho units): '
READ(*,*) RA
WRITE(*,*) 'ENTER OUTER CYLINDER RADIUS (ko*rho units): '
READ(*,*) RP
WRITE(*,*) 'ENTER NUMBER OF NODES'
READ(*,*) NODES
WRITE(*,*) 'ENTER NUMBER OF INTEGRATION SEGMENTS: '
READ(*,*) NSEG
WRITE(*,*) 'ENTER FREQUENCY (Hz): '
READ(*,*) FREQ
C
SUM=0.0
LAMBDA=C/FREQ
```

```

K0=2*PI/LAMBDA
ARCLN=2*PI*RA/NSEG
DPHI=2.0*PI/(NODES-1.0)
DPHIP=2.0*PI/NSEG
DO 2 L=1,NODES-1
  FTOT=(0.0,0.0)
  PHI=-(L-1)*DPHI
  DO 1 I=0,NSEG-1
    PHIP=-(I-0.5)*DPHIP
    CP=COS(PHIP-PHI)
    PSI=EXP(-J*RA*CP)
    DPSI=J*K0*CP*PSI
C   THIS IS ACTUALLY KoR
    R=SQRT(RP*RP+RA*RA-2.*RP*RA*CP)
    CA=(RA-RP*CP)/R
    RC=CMPLX(R)
    CALL HAN1(RC,HH0,HH1)
    F=((J/4*HH0) * DPSI) - (PSI * (-J*K0/4) * HH1 * CA)
    FTOT=FTOT+F
1  CONTINUE
    INT=FTOT*ARCLN
    DEG=360. + (PHI*180/PI)
    WRITE(10,22) DEG,CABS(INT)
    SUM=SUM+CABS(INT)
2  CONTINUE
    PSIAVG=SUM/(NODES-1)
    WRITE(10,*) 'PSIAVG =',PSIAVG
22 FORMAT(F7.1,4X,F9.6)
STOP
END

```

SUBROUTINE HAN1(Z,H0,H1)

```

C
C   Computing Hankel Functions for n=0,1 with
C   Complex Argument, Z.   Direct Power Series Method for
C   CABS(Z) .LE. 5   and Hankel's Asymptotic Formula for
C   CABS(Z) .GT. 5.   Written 11/6/87 by M.A. Morgan
C

```

```

INTEGER M,M2
REAL C(34),DM,F(34),G0,P(34),Pi,P2
COMPLEX Z,Z2,Z3,Z4,J0,J1,Y0,Y1,AM,CL,P0,P1,Q0,Q1
COMPLEX E0,E1,X0,X1,H0,H1,j
PI=3.1415927
P2=2.0/PI
j=(0.,1.)
IF(CABS(Z).LE.5.0) THEN

```

```

C
C   Direct Power Series Method
C

```

```

      G0= 1.78072
      Z2=0.5*Z
      CL=CLOG(G0*Z2)
C
C   Computing F(m) = m ! and P(m) = 1 + 1/2 + 1/3 + ....+ 1/m
C
      F(1)=1.0
      P(1)=1.0
      DO 11 M=2,34
          F(M)=M*F(M-1)
          P(M)=P(M-1)+1.0/M
11      CONTINUE
C
C   Computing Power Series Coefficients
C
      DM=-1.0
      DO 22 M=1,34
          C(M)=DM/(F(M)*F(M))
          DM=-DM
22      CONTINUE
C
C   Computing J0 and J1
C
      J0=(1.,0.)
      J1=(0.,0.)
      M=0
33      M=M+1
      M2=2*M
      AM=C(M)*(Z2**M2)
      J0=J0+AM
      J1=J1-M*AM
      IF((CABS(AM).GT.1.0E-10).AND.(M.LT.34)) GO TO 33
      J1=J1/Z2
C
C   Computing Y0 and Y1
C
      M=0
      Y0=CL*J0
      Y1=Z2*CL*J1-0.5*J0
44      M=M+1
      M2=2*M
      AM=C(M)*P(M)*(Z2**M2)
      Y0=Y0-AM
      Y1=Y1+M*AM
      IF((CABS(AM).GT.1.0E-10).AND.(M.LT.34)) GO TO 44
      Y0=P2*Y0
      Y1=P2*Y1/Z2
      H0=J0-j*Y0
      H1=J1-j*Y1
      RETURN

```

```

ELSE
C
C Hankel' Asymptotic Formula (Abram. & Stegun p. 364)
C
      Z2=Z*Z
      Z3=Z*Z2
      Z4=Z*Z3
      P0=1.0-.0703125/Z2+.1121521/Z4
      Q0=-.125/Z+.0732422/Z3
      P1=1.0+.1171875/Z2-.1441956/Z4
      Q1=.375/Z-.10253906/Z3
      X0=(Z-.25*PI)
      X1=(Z-.75*PI)
      E0=CEXP(-j*X0)
      E1=CEXP(-j*X1)
      AM=CSQRT(P2/Z)
      H0=AM*(P0-j*Q0)*E0
      H1=AM*(P1-j*Q1)*E1
ENDIF
C
RETURN
END

```

LIST OF REFERENCES

1. J. W. Crispin and K. M. Siegel, *Methods of Radar Cross-Section Analysis*, Academic Press, New York, 1968
2. Jack H. Richmond, "Scattering by a dielectric cylinder of arbitrary cross section shape," *IEEE Transactions on Antennas and Propagation*, Vol. AP-13, May 1965, pp. 334-341.
3. Constantine A. Balanis, *Advanced Electromagnetic Engineering*, John Wiley and Sons, New York, 1989.
4. Mogens G. Andreasen, "Scattering from cylinders with arbitrary surface impedance," *Proceedings of the IEEE*, August 1965, pp. 812-817.
5. J. Van Bladel, *Electromagnetic Fields*, Hemisphere Publishing Company, New York, 1985.
6. M. A. Morgan, "Principles of Finite Methods in Electromagnetic Scattering," in *Finite Element and Finite Difference Methods in Electromagnetic Scattering*, Michael A. Morgan, ed., Elsevier, New York, 1990.
7. R. Janaswamy, Unpublished notes, Naval Postgraduate School, Monterey, California, dated January 1990.
8. William H. Press and others, *Numerical Recipes*, Cambridge University Press, Cambridge, 1986.
9. Carl de Boor, "Cadre: An Algorithm for Numerical Quadrature," in *Mathematical Software*, John R. Rice, ed., Academic Press, New York, 1971.
10. J. J. Bowman and others, eds., *Electromagnetic and Acoustic Scattering by Simple Shapes*, North-Holland Publishing Company, Amsterdam, 1969.
11. Thaddeus B. Welch III, "Electromagnetic scattering from two-dimensional objects using the field feedback formulation," Master's thesis, Naval Postgraduate School, Monterey, California, 1989.